# Superstructure Optimization of Petroleum Refinery Design: Processing Alternatives for Naphtha Produced from the Atmospheric Distillation Unit

by

Yeoh Xiao Qi

Dissertation submitted in partial fulfillment of the requirements for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

APRIL 2009

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

# Superstructure Optimization of Petroleum Refinery Design: Processing Alternatives for Naphtha Produced from the Atmospheric Distillation Unit

by

Yeoh Xiao Qi

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Chemical Engineering)

Approved by,

C. KHOR

**(MR. KHOR CHENG SEONG)**

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

April 2009

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

(YEOH XIAO QI)

# ABSTRACT

This research project concerns superstructure optimization for the design of petroleum refineries focusing on the subsystem that considers the alternatives for naphtha produced from the atmospheric distillation unit (ADU). The intricate complexities associated with this process synthesis problem in general and the refinery design problem in specific necessitates the development and implementation of a systematic and automated approach that efficiently and rigorously integrate the elaborate interactions involving the design decision variables. The primary objective of this research is to establish a systematic procedure to determine the optimal topology of the refinery subsystem of naphtha produced from the ADU using the optimization or mathematical programming approach. Through the identification of equipment, raw materials, products, and process alternatives in terms of the different feasible choices of states (material streams) and tasks (process units) for the mentioned subsystem, the first step is to represent the problem as the interconnections between these elements in a network representation of a superstructure. Subsequently, an optimization model is formulated with binary and continuous variables in order to arrive at the optimum flowsheet design. The scope of this work is focused on the formulation of a mixed-integer linear programming (MILP) and a generalized disjunctive programming (GDP) optimization models. The independent design decision variables are flows of materials and the continuous variables of stream flowrates with the discrete variables denoting the existence of streams. Logical constraints are extensively incorporated in the models to represent qualitative design knowledge through design specifications and structural specifications on the interconnectivity relationships involving the states and the tasks. Computational studies to demonstrate the implementation of the proposed modeling approaches are carried out on the GAMS modeling language platform using the established GAMS/CPLEX solver and the new code of GAMS/LOGMIP solver for the MILP and GDP, respectively. Two design scenarios are considered as distinguished by the API gravity (specific gravity) of the crude charge to the ADU. The optimal refinery topology generated from the MILP and GDP model agree with the typical existing refinery topology. The way forward for this project is to account for varying sulphur content in the crude charge as well as to introduce nonlinearity in the composition modeling to obtain a more practical representation of a real-world refinery design problem.

## ACKNOWLEDGEMENTS

I would like to take this opportunity to thank everyone whom had given their support and help throughout the whole period of completing this project.

First and foremost, I would like acknowledge the endless help and support received from my supervisor, Mr. Khor Cheng Seong throughout the whole period of completing this final year project. His guidance has really been the main source of motivation and has driven me in completing this project successfully.

Finally, I would like to thank all my fellow colleagues for their assistance and ideas in completion of this project.

(YEOH XIAO QI)

## LIST OF FIGURES

## LIST OF TABLES

# CHAPTER 1
# INTRODUCTION

## 1.1 BACKGROUND OF STUDY

The goal of optimization is to find the values of the variables in the process that yield the best value of the performance criterion. Linear programming (LP) has been around since the 1940s and has now reached a very high level of advancement with the meteoric rise in computing power.

Optimization is the core objective of chemical process design. Selecting the best among a set of possible solutions requires good engineering judgment to critically analyze the process with respect to the desired performance objectives. It is crucial to identify and strike a balance between the competing objectives of realizing the largest production, the greatest profit, the minimum cost, the least energy usage, and so on. This ensures improved plant performance through improved yields of valuable products, higher processing rates, longer time between shutdowns and reduced maintenance costs. In order to find the best solution within the given constraints and flexibilities, a trade-off usually exists between capital and operating costs.

Even though the design stage only takes up about 2 or 3 percent of the project expenditure, decisions made during this phase have an immense impact on plant economic performance because approximately 80 percent of the capital and operating expenses of the final plant are fixed during the design stage (Biegler et al, 1997, p. xviii). Hence, the necessity of developing systematic methods in chemical process design has led to two approaches to process synthesis: hierarchical decomposition and mathematical programming.

In process synthesis, there are two major approaches to determine the optimal configuration of a flowsheet and its operating condition. First approach, the problem can be solved in sequential form, by decomposition, fixing some elements in the flowsheet, and then using heuristic rules to determine changes in the flowsheet that may lead to an improved solution. An example of such an approach is the sequential hierarchical decomposition strategy by Douglas (1988). The sequential nature of the decisions and the heuristic rules that are used can lead to sub-optimal designs. Douglas (1998) claims that only 1 percent of all designs are ever implemented in practice and hence this screening procedure avoids meticulous evaluation of most alternatives. It is not possible to rigorously produce an optimal design because the sequential nature of flowsheet synthesis cannot take all interactions among the design variables into consideration. Furthermore, the exponential number of possible topologies coupled with multitude of process technology options decreases the chances of realizing the best design.

The second strategy that can be applied to solve a process synthesis problem is based on simultaneous optimization using mathematical programming (Grossmann, 1996). This strategy requires postulation of a superstructure that includes equipment that can be potentially selected in the final flowsheet, as well as their interconnection. The equations of the equipment and their connectivity, and constraints for the operating conditions are then incorporated in an optimization problem where an objective function is specified such as cost minimization or profit maximization. This approach requires the discrete variables to represent the choices of equipment, with which the model becomes a mixed integer linear or non-linear program (MILP or MINLP). (Grossmann, 1996) states that the advantage of mathematical programming strategies is that they perform simultaneous optimization of the configuration and operating conditions. The drawback is that global optimality conditions cannot be guaranteed for nonlinear models unless specific methods for global optimization are used.

The complexities involved in designing a petroleum refinery are numerous and difficult. Many factors such as design specification and structural specification constraints have to be considered and incorporated in the conceptual design to arrive at an optimum configuration or topology of the refinery flowsheet. Hierarchical decomposition utilizes heuristics, short-cut design procedures and engineering experience are usually adopted to develop an initial base-case. However, these approaches are possibly time-consuming and the end result does not guarantee the desired optimal decision. Thus, the development of an automated systematic procedure in the refinery design endeavour will significantly improve the decisionmaking process. This can be achieved via the optimization or mathematical programming approach by representing the problem through a superstructure and formulating the corresponding mathematical model, which is solved to obtain the optimal refinery topology based on the input of crude oil to be processed and the output of final refining products as dictated by market demands.

## 1.2 RESEARCH MOTIVATION

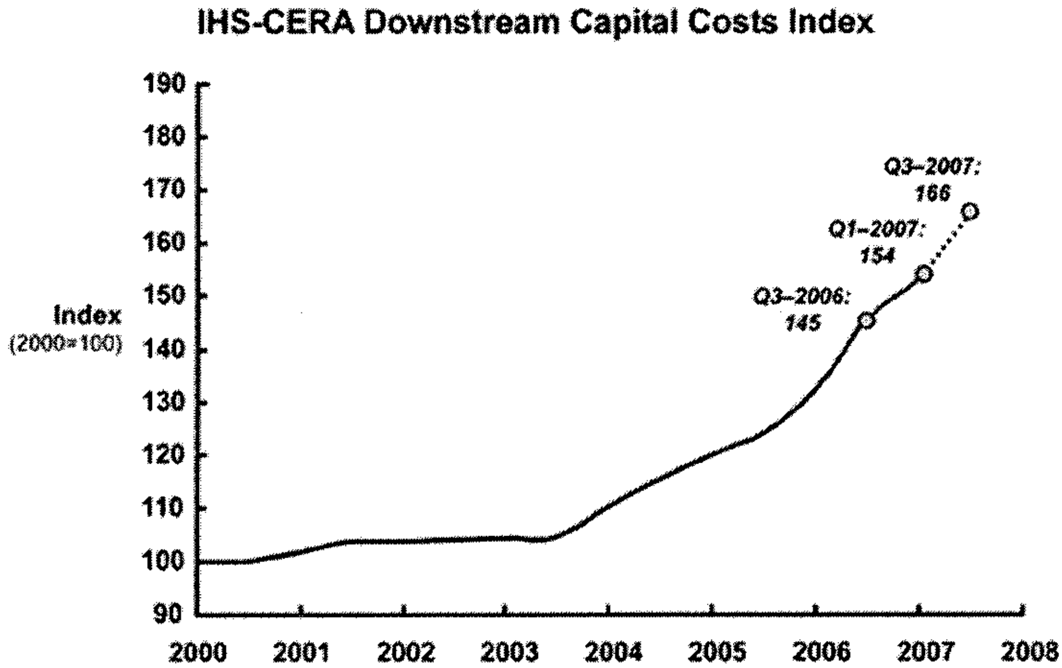


**Figure 1.1** Downstream capital cost index

(Cambridge Energy Research Associates, 2007)

Figure 1.1 shows the rapid increasing downstream capital cost index from middle of year 2003 to year 2008. Automated approach that guarantees optimal refinery design is increasingly important due to increase in capital costs, higher energy costs, depleting energy sources. The rising consumption of fuel has led to a higher demand for petroleum products despite tight supplies, have witnessed the call for the construction of new grassroots petroleum refineries in countries notably the US (such as in the states of Arizona and Louisiana) and also in the Middle East countries. Consequently, consumer demand provided the incentive for the construction of new refineries.

However, with increasingly strict environmental regulations and emphasis on clean fuels, new refineries will have narrower operating margins and more stringent product specifications to adhere to. This adds to the degree of complexity in designing refineries, which at present is already time-consuming, with the intricacies of the interplay among environmental factors, public opinions, and a host of time-consuming permitting processes. This has given rise to an exponential number of possible refinery topologies or configurations that adequately meets current economic, operating, and environmental requirements.

Hence it is even more crucial to develop and implement automated approaches in optimizing process design to efficiently integrate current economics as well as operating and environmental requirements to arrive at an optimum configuration or topology of the refinery flowsheet. Clearly, there is substantial motivation for the development and implementation of systematic and automated approaches and methods for designing refineries. Within the realm of chemical engineering, the problem is generalized as a process synthesis/conceptual process design problem.

## 1.3 PROBLEM STATEMENT

The problem statement can be addressed as: given is a set of equipment, raw materials, products and process alternatives in terms of different choices of tasks and equipment, and the interconnections among them, the author is to establish a systematic procedure for representing these elements in a superstructure, and deriving a mathematical programming model with discrete and continuous variables to predict an optimum flowsheet design. This research project concerns the optimal design of the topology or configuration of a refinery that addresses the following aspects:

- the discrete decisions involving: (1) the **selection** of the process units (tasks) and material streams (states); and (2) the **sequence** of the interconnections among the units and the streams; and
- the continuous decisions involving the optimal design flowrates of the streams.

This research project will be undertaken using the optimization approach of mixed integer linear programming (MILP) and generalized disjunctive programming (GDP) concerning only flows of material.

## 1.4 OBJECTIVES AND SCOPE OF STUDY

The objectives of this research work are as follows:

1. To develop a superstructure representation for a refinery network topology with a suitable level of detail by considering the processing alternatives for naphtha;
2. To develop optimization models based on the superstructure representation by adopting two formulations: (1) the mixed-integer linear programming (MILP) framework and (2) the generalized disjunctive programming (GDP) framework, incorporating both continuous and discrete decisions. The model formulation consists mainly of: (a) constant-yield-based linear material balances, and (b) logical constraints enforcing the design specifications and structural specifications, in which the latter stipulates the interconnectivity relationships among the units and the streams for the selection and sequencing of alternative processing routes;
3. To solve the MILP optimization model using Cplex solver and GDP formulation using LOGMIP solver within the GAMS modeling language environment;
4. To analyze and compare the MILP model formulation against that of the GDP.

# CHAPTER 2
# LITERATURE REVIEW

The optimization approach in process synthesis consists of three main procedures: representation of alternatives, mathematical modeling and algorithmic development.

## 2.1 SUPERSTRUCTURE REPRESENTATION OF DESIGN ALTERNATEIVES

A superstructure embeds all feasible process design alternatives of interest by incorporating the different competitive process units and their possible interconnections. Hence, each alternative can be a feasible or optimal process flowsheet.

According to systematic modeling framework of superstructure optimization in process synthesis by Yeomans and Grossmann (1999), there are three basic nodes in a superstructure: states, tasks and equipment. States are a set of physical and chemical properties that identify a stream in a process, denoted by circles, which represents the feeds, intermediates, and final products. Examples include temperature, pressure, mass flow and composition of a stream. Tasks are the physical and chemical transformations of process operations that correspond to momentum, mass and energy transfer operations such as reaction, absorption and mixing. Task nodes represent the process operations that transform material from one or more input states into one or more output states, denoted by rectangles. Equipment is the physical devices that execute a given task, such as reactor, heat exchanger, distillation column, heat exchanger, and absorber.

There are three types of superstructure representation, namely: State-Task Network (STN), State-Equipment Network (SEN) and Resource-Task Network (RTN).

### 2.1.1 State–Task Network (STN) Superstructure Representation

STN assumes that processing tasks produce and consume states. The states and tasks are defined first, leaving unknown equipment assignment to a second stage. Feedstock must be connected to product and vice-versa. Each intermediate state and task must be on at least one such path. Some tasks are conditional; others must be present in all design alternatives. There is no need to distinguish one from the other at the level of representation, but only at the level of model. One or more of these operations (temperature, momentum, mass or energy transfer) may be performed in one task if technically feasible.

When the states and the tasks are identified, the equipment assignment can be carried out in two ways: One-Task-One-Equipment (OTOE) assignment or the Variable-Task-Equipment (VTE) assignment. In OTOE, each task is distinguished from another and no two equipments are assigned to the same task. If a task can be executed by two different equipments, the tasks will have to be redefined to distinguish one from the other. Equipment assignment is explicitly performed, in which case, the representation by both task and equipment is identical. In (Yeomans and Grossmann, 1999), the advantages of OTOE are (1) it is most straightforward representation from which a clear optimization model can best be formulated; (2) it handles the assignment of equipment implicitly by reducing the combinatorial complexity of the mathematical model

In Yeomans and Grossmann (1999), VTE enables a single equipment to be assigned different tasks and vice versa. A set of equipment that can perform all the tasks needed is identified first. Assignment of equipment to task is part of the optimization model. A single equipment unit can be assigned to different tasks and a single task can be assigned to different equipment. It is necessary to select one and only one of the equipment configurations available for the task

### 2.1.2 State–Equipment Network (SEN) Superstructure Representation

SEN considers full connectivity among states and equipment. It defines the tasks and equipment, leaving the assignment of tasks to equipment to be determined. It includes different states of process and equipment that are likely to be used. One or more different tasks which can be performed by single equipment are determined by considering full connectivity between states and equipment. Tasks that can take place in specific equipment are not pre-specified. Assignment of task to equipment is part of the optimization model (task assignment is an unknown) which is equivalent to VTE of STN. SEN leads to a smaller combinatorial problem for equipment selection but implicit combinatorial complexity is present in the possible equipment interconnections. Also, state definition is not unique and all possible realizations of the streams that will originate from a certain task will have to be taken into account. This can complicate the modeling stage. The SEN representation is also useful for retrofit design problems as it shows explicitly the exiting equipment. (Yeomans and Grossmann, 1999)

### 2.1.3 Resource–Task Network (RTN) Superstructure Representation

RTN assumes that a task only consumes and produces resources. The concept of resource includes all entities that are involved in the process steps, such as materials (raw-materials, intermediates and products), processing and storage equipment (tanks, reactors, etc.), utilities (operators, steam, etc.) as well as equipment conditions (clean, dirty). There is no distinction between equipment of any type and other resources. All resources are allowed to be produced or consumed by the tasks at any time during their execution (processing items are treated as though it is consumed at the start of a task and produced at the end). Circles denote states as well as other resources (processing units and vessels). RTN formulation is able to capture additional features of a problem in a straightforward manner, giving it the advantage of conceptual simplicity and direct applicability to a large number of complex process scheduling problems. (Barbosa-Povoa and Pantelides, 1997)

For the purpose of this research, the superstructure is constructed using STN representation or more specifically, using OTOE representation. This is in accordance with the findings reported by Yeomans and Grossmann (1999), which says:

1. OTOE is the most straightforward representation from which a clear optimization model can best be formulated to establish a systematic approach for determining the optimum topology for the refinery subsystem of the naphtha produced from the Atmospheric Distillation Unit;
2. OTOE handles the assignment of equipment implicitly, therefore reducing the combinatorial complexity of the mathematical model.

## 2.2 SUPERSTRUCTURE OPTIMIZATION MODELING

The objective of optimization problem is to find the optimized objective function (profit maximization, cost minimization, environmental impacts) by specifying values of the variables that satisfy equality and inequality constraints, thus to yield an optimal solution. Developing models to account for these constraints is closely related to the resolution of the optimization model. Mathematical programming can accommodate models of various degree of complexity, which can be classified into three main classes (Grossmann et al, 1999):

1. *Aggregated models* - These refer to high level representations in which the design or synthesis problem is greatly simplified by an aspect or objective that tends to dominate the problem at hand.
2. *Short cut models* - These refer to fairly detailed superstructures that involve cost optimization (investment and operating costs), but in which the performance of the units is predicted with relatively simple nonlinear models in order to reduce the computational cost, and/or for exploiting the algebraic structure of the equations, especially for global optimization.
3. *Rigorous models* - These also rely on detailed superstructures, but involve rigorous and complex models for predicting the performance of the units. The area of synthesis of distillation sequences (ideal and non-ideal) is perhaps the one that has received the most attention for developing rigorous models.

Aggregated models yield simpler types of optimization models, namely Linear Programming (LP), Non-Linear Programming (NLP) or Mixed-Integer Linear Programming (MILP) models. These are easier to solve compared to Mixed-Integer Non-Linear Programming (MINLP) models, which are the usual result for both short cut and rigorous models. Hence, the model formulated in this research is an aggregated model. If all functions are linear, it corresponds to a MILP as used in the modeling of this research. (Floudas, 1995).

The advantages of MILP include enable qualitative design knowledge to be included in design and synthesis problems using logical constraints on design & structural specs. The qualitative design information (e.g., engineering experience, heuristics) plays important role in process design & synthesis which concern decisions on process units to be integrated in a flowsheet of process network. Example is to influence decisions on selection of treatment and conversion technologies for crude oil processing. Also Raman and Grossmann (1991) demonstrate how both heuristics and logic relationships describing connections and interactions among the process units in a superstructure, which are expressed in the form of propositional logics, can be represented in terms of linear inequalities via binary variables.

## 2.3 RELATION BETWEEN LOGICAL INFERENCE & MIXED-INTEGER LINEAR PROGRAMMING (MILP) MODELING

In order to obtain an equivalent mathematical representation for any propositional logic expression, one must first consider basic logical operators to determine how each can be transformed into an equivalent representation in the form of an equation or inequality. These transformations are then used to convert general logical expressions into an equivalent mathematical representation.

The basic unit of propositional logic expression, which can correspond to a state or to an action, is called a literal which is a single variable that can assume either of two values, true or false. Associated with each literal P, there is another literal NOT *P* $(\neg P)$ such that either *P* or $(\neg P)$ is always true. A clause is a set of literals separated by OR operators and is also called a disjunction. A proposition is any logical expression and consists of a set of clauses *Pi*, *i*=1 are related by the logical operators OR, AND, IMPLICATION, as stated in Raman and Grossmann (1991).

To each proposition $P_i$, a binary variable $y_i$ is assigned. Then the negation or complement of $P_i$,$(\neg P_i)$ is given by $1 - y_i$. The logical value of true corresponds to the binary value of 1 and false corresponds to the binary value of 0. The basic operators used in propositional logic and the representation of their relationships are shown in Table 1. The procedure to convert a logical expression into its corresponding conjunctive normal form was formalized by Clocksin & Mellish (1981).

The three steps procedures to transform each logical proposition are:

1) replace the implication by its equivalent disjunction:

   $P_1 \Rightarrow P_2 \Leftrightarrow \neg P_1 \vee P_2$ (1)

2) move the negation inward by applying DeMorgan's Theorem:

   $\neg(P_1 \wedge P_2) \Leftrightarrow \neg P_1 \vee \neg P_2$ (2) $\neg(P_1 \vee P_2) \Leftrightarrow \neg P_1 \wedge \neg P_2$ (3)

3) recursively distribute the "OR' over the "AND":

   $(P_1 \wedge P_2) \vee P_3 \Leftrightarrow (P_1 \vee P_2) \wedge (P_1 \vee P_3)$ (4)

Having converted each logical proposition into its conjunctive normal form representation, $Q_1 \wedge Q_2 \wedge ... \wedge Q_S$, it can then be easily expressed as a set of linear equality and inequality constraints.

**Table 2.1:** Representation of logical relations with linear inequalities (Raman and Grossmann (1994))

| **Logical operator** | **Example of use for process networks** | **Logic proposition** | **Logical Boolean expression** | **Representation as algebraic integer linear inequality/equality constraint** |
|---|---|---|---|---|
| Logical OR | For selection of at least one process unit (or more than one unit or all of the units) in consideration | --- | $P_1 \vee P_2 \vee L \ \vee P_r$ | $y_1 + y_2 + L \ + y_r \geq 1$ |
| Logical AND | For selection of all process units in consideration | --- | $P_1 \wedge P_2 \wedge L \ \wedge P_r$ | $y_1 \geq 1; y_2 \geq 1; L \ ; y_r \geq 1$ |
| Implication | Select unit 1 only if unit 2 is selected (e.g., select FCC only if the upstream HDS is selected) | $P_2$ only if $P_1$<br>$P_1 \Rightarrow P_2$ is logically equivalent to $\neg P_1 \vee P_2$ | $\neg P_1 \vee P_2$ | $(1-y_1)+y_2 \geq 1$<br>$y_1 - y_2 \leq 0$ or $y_1 \leq y_2$ |
| Equivalence | Selecting a unit implies the selection of another unit or other units | $P_1$ if and only if $P_2$<br>$(P_1 \Rightarrow P_2) \wedge (P_2 \Rightarrow P_1)$<br>which can also be written as: $P_1 \Leftrightarrow P_2$ | $(\neg P_1 \vee P_2) \wedge (\neg P_2 \vee P_1)$ | $(1-y_1)+y_2 \geq 1$ and $(1-y_2)+y_1 \geq 1$<br>$-y_1+y_2 \geq 0$ and $-y_2+y_1 \geq 0$<br>$y_1-y_2 \leq 0$ and $y_2-y_1 \leq 0$<br>$y_1 \leq y_2$ and $y_2 \leq y_1$<br>or<br>$y_1 = y_2$ |
| Exclusive OR (EOR) | For selection of only one process unit (or material stream) | Exactly one of the variables is true | $P_1 \veebar P_2 \veebar L \ \veebar P_r$ or this can equivalently be written as:<br>$P_1$ EOR $P_2$ EOR ...EOR $P_r$ | $y_1 + y_2 + L \ + y_r = 1$ |
| Classification | For selection of any process unit. | $Q = \{P_1, P_2, \ldots, P_r\}$<br>$Q$ is true if any of the variables inside the brackets are true | --- | $y_q = y_1 + y_2 + L \ + y_r$ |
| "Combination" of Equivalence and OR | Selection of at least one process unit (or more than one unit or all of the units) implies the selection of another unit or other units | --- | $(P_1 \vee P_2) \Leftrightarrow P_3$ | $p_3 \geq p_1$<br>$p_3 \geq p_2$<br>$p_1 + p_2 \geq p_3$ |
| "Combination" of Equivalence and EOR | Selection of only one process unit (or material stream) implies the selection of another unit or other units | --- | $(P_1 \veebar P_2) \Leftrightarrow P_3$ | $p_1 + p_2 = p_3$ |

## 2.4 GENERALIZED DISJUNCTIVE PROGRAMMING (GDP) MODELING

The general framework of superstructure optimization in process synthesis is composed of three major steps which can in principle be applied to any synthesis problem to derive a mathematical programming model for predicting an optimal flowsheet configuration.

The initial step is to consider two major superstructure representations: the STN (State-Task-Network), in which the tasks and states are defined while the equipment assignment is generally unknown; and the SEN (State-Equipment-Network) in which the tasks and the equipment are defined while the assignment of tasks to equipment must be determined. Based on these network representations, the corresponding synthesis problems will be modeled with GDP. These logic based methods will then be used as basis for deriving algebraic mixed-integer optimization models.

GDP modeling is the second step of the proposed framework for process synthesis corresponds to the modeling of the chosen representation, STN or SEN, as a mathematical programming problem. Since there will be conditional tasks or equipment that might be selected or not in the final flowsheet, it is necessary to use a discrete mathematical programming model. The use of disjunctive programming is of particular interest, since process synthesis problems naturally lead to models where the solution space is disjoint, and there is a strong logic on the connectivity among the different tasks (Raman & Grossmann 1993, 1994).

In order to use GDP (Raman & Grossmann, 1994) to model the STN or SEN representations, it is necessary to identify the conditional constraints from among those that must hold for all synthesis alternatives. The conditional constraints will be represented with disjunctions and assigned a Boolean variable that represents its existence (if the Boolean variable takes a value of 'true'). In general mixers and splitters can be considered conditional tasks. However, if the equations that are applied to the mixer and splitter are only mass and energy balances, these constraints do not involve any type of discrete decision or discrete variable assignment for them to be valid.

According to Yeomans and Grossmann (1999), the advantages of GDP are: (1) allows a symbolic/quantitative representation of discrete and continuous decisions in design problems; (2) avoids the use of big-M logical constraints which yields a poor relaxation; (3) no binary $0-1$ variables are explicitly included in the model; (4) reduces problem size by only considering disjunctions for which Boolean variable is true; (5) reduces combinatorial search effort in which less combination of binary variables to be evaluated.

In Turkay and Grossmann (1995), the GDP hybrid modeling formulation is given by:

$$\min Z = \sum_i c_i + f(x) + d^T y \qquad (5)$$

$$s.t.$$

$$\left.\begin{array}{l} g(x) \leq 0 \\ r(x) + D(y) \leq 0 \\ Ay \geq 0 \end{array}\right\} \qquad (6)$$

$$\begin{bmatrix} Y_i \\ h_i(x) \leq 0 \\ c_i = \gamma_i \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ B^i x = 0 \\ c_i = 0 \end{bmatrix} \quad i \in D \qquad (7)$$

$$\Omega(Y) = True \qquad (8)$$

$$x \in R^n, c_i \geq 0, \mathrm{Y} \in \{\text{True, False}\}^m \qquad (9)$$

The model includes disjunctions, binary variables and integer or mixed-integer constraints. The nonlinear model (7) involved three types of variables: $x$ (flow, pressure, temperatures) and $c_i$ (fixed charge) are the continuous variables; the Boolean variable, Yi shows the existence of units and to indicate whether a given disjunction $i$ is true or false. The set of disjunctions, D, apply for the processing unites. If a process unit exists, ($Y_i$ = True), then the equations and constraints describing that unit are enforced and a fixed charge is applied; else ($\neg Y_i$ = False), a subset of continuous variables and the fixed charge are set to zero.

The main advantages of generalized disjunctive programs in structural flow-sheet optimization are its robustness and computational efficiency when compared to algebraic MILP models and algorithms. (Vecchietti and Grossmann, 1998) Using disjunction in the problem formulation is preferred to avoid using of big-$M$ constraints which yield poor relaxation and prevent zero flows.

In general, at least three approaches are available to solve GDP:

1. reformulation of the disjunctions in GDP into MILP via big-$M$ reformulation;
2. reformulation of the disjunctions in GDP into MILP via convex hull formulation, which provides tighter relaxation compared to the first approach as according to Turkay and Grossmann (1996));
3. solution of GDP using GAMS/LOGMIP solver.

## 2.5 GENERALIZED DISJUNCTIVE PROGRAMMING (GDP) MODELS FOR STATE-TASK NETWORK (STN) REPRESENTATION

The following sets and variables must be defined to formulate the GDP model with direct formulation on process unit existence. Let $t \in T$ define the set of tasks in the superstructure, where $T = T_P \cup T_C$ and $T_P$ is the set of permanent tasks and $T_C$ is the set of conditional tasks that may be selected. Let $s \in S$ defines the set of states, and $j \in E$ defines the set of equipment units. Let $I_t = \{s \mid s$ is an input state of task $t\}$, and $O_t = \{s' \mid s'$ is an output state of task $t\}$. The variables $z_t$, $x_s$ and $d_j$ are used to represent the operating variables in the tasks, the flow and state variables interconnecting the states, and the design variables for the equipment, respectively. The function $h_t(z_t, x_s, x_{s'})$ represents the equations (mass balances, energy balances, etc.) and constraints corresponding to task $t$, and $r_j(d_j, x_s, x_{s'}, z_t)$ represents the equations and constraints corresponding to a particular equipment design. Finally, $f(d_j, z_t)$ represents the cost function in terms of the design and control variables, $d_j$ and $z_t$. (Yeomans and Grossmann, 1999)

If the OTOE case is considered for the STN superstructure, the equations and constraints from equipment and tasks can be integrated in vector $g_t = [\, h_t(z_t, x_s, x_{s'}), r_j(d_{j'}, x_s, x_{s'}, z_t)]^T$ where $j' \in Q_t = \{ j' \in E \mid j'$ is associated with task $t\}$, and $\cap t \in TQ_t = \varnothing$. The GDP model for STN/OTOE representation is then as follows:

$$(\text{P-STN1}): \ \min \sum_{t \in T} c_t + \sum_{s \in S} \alpha_s x_s \qquad (10)$$

$$\left.\begin{array}{l} s.t\ g_t\ (d_j, z_t, x_s, x_{s'}) \le 0 \\ c_t = f(d_j, z_t) \end{array}\right\} \begin{array}{ll} j' \in Q_{t'} & t \in T_P \\ s \in I_t & s' \in O_t \end{array} \qquad (11)$$

$$\begin{bmatrix} Y_t \\ \left.\begin{array}{l} g_t\ (d_j, z_t, x_s, x_{s'}) \le 0 \\ c_t = f(d_j, z_t) \end{array}\right\} \begin{array}{ll} j' \in Q_{t'} & t \in T_P \\ s \in I_t & s' \in O_t \end{array} \end{bmatrix} \vee \begin{bmatrix} \neg Y_t \\ \left.\begin{array}{l} d_{j'} = z_t = 0 \\ x_s = x_{s'} = 0 \end{array}\right\} \begin{array}{ll} j' \in Q_t & \\ \forall s \in I_t & s' \in O_t \end{array} \end{bmatrix} t \in T_c \qquad (12)$$

$$\Omega(y) = True \qquad (13)$$

$$d \in D, z \in Z, x \in X \quad Y_t = \{\text{True, False}\}$$

Eq (10) represents the objective function which is the cost incurred by the selection of a task with its equipment and variable costs. Eq (11) represents the mass and energy balances, and the design constraints. Eq (12) represents the selection of a conditional task $t \in T_C$ is represented by a Boolean variable. Eq (13) represents the logic relations between Boolean variables (PSTN-1)

## 2.6 GAMS/LOGMIP AS A DISJUNTIVE BINARY LINEAR SOLVER FOR PROCESS SYSTEM MODEL

According to Vecchietti and Grossmann (1999), LOGMIP is a computer code written in C allowing the specifications of disjunctions in the problem formulation. The program code is written in C to assure portability to other platforms. LOGMIP is layered over the GAMS modeling language in which the GAMS input-output library has been used to link LOGMIP to GAMS. LOGMIP can provide a rather general modeling framework and solution tool for solving disjunctive, algebraic work or hybrid linear optimization problem.

The GAMS modeling language is used to write the model in terms of disjunctions and algebraic equations; also it is used for specifying the disjunctions by controlling the domain of definitions using a dollar sign. The logic relations between the Boolean variables are handled as inequalities in LOGMIP. PROLOG is used to transform the logical propositional expression into equivalent mathematical linear form. LOGMIP has a model recognition which can identify different types of model form as discussed in Table 2.2.

**Table 2.2**: LOGMIP model formulation type

| | |
|---|---|
| **Hybrid model formulation**<br>▪ involves disjunctions, binary variables, and Boolean variables | $\min Z = \sum_i c_i + f(x) + d^T y$<br>$s.t.$<br>$g(x) \le 0$<br>$r(x) + D(y) \le 0$<br>$Ay \ge 0$<br>$\begin{bmatrix} Y_i \\ h_i(x) \le 0 \\ c_i = \gamma_i \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ B^i x = 0 \\ c_i = 0 \end{bmatrix} \quad i \in D$<br>$\Omega(Y) = True$<br>$x \in R^n, c_i \ge 0, Y \in \{\text{True, False}\}^m$ |
| **Algebraic model**<br>▪ involves binary variables | $\min Z = f(x) + d^T y$<br>$s.t.$<br>$g(x) \le 0$<br>$r(x) + D(y) \le 0$<br>$Ay \ge 0$<br>$x \in R^n, y \in \{\text{True, False}\}^m$ |
| **Disjunctive representation**<br>▪ involves disjunctions and Boolean variables | $\min Z = \sum_i c_i + f(x)$<br>$s.t.$<br>$g(x) \le 0$<br>$\begin{bmatrix} Y_i \\ h_i(x) \le 0 \\ c_i = \gamma_i \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ B^i x = 0 \\ c_i = 0 \end{bmatrix} \quad i \in D$<br>$\Omega(Y) = True$<br>$x \in R^n, c_i \ge 0, Y \in \{\text{True, False}\}^m$ |

## 2.7 REFINING PROCESS

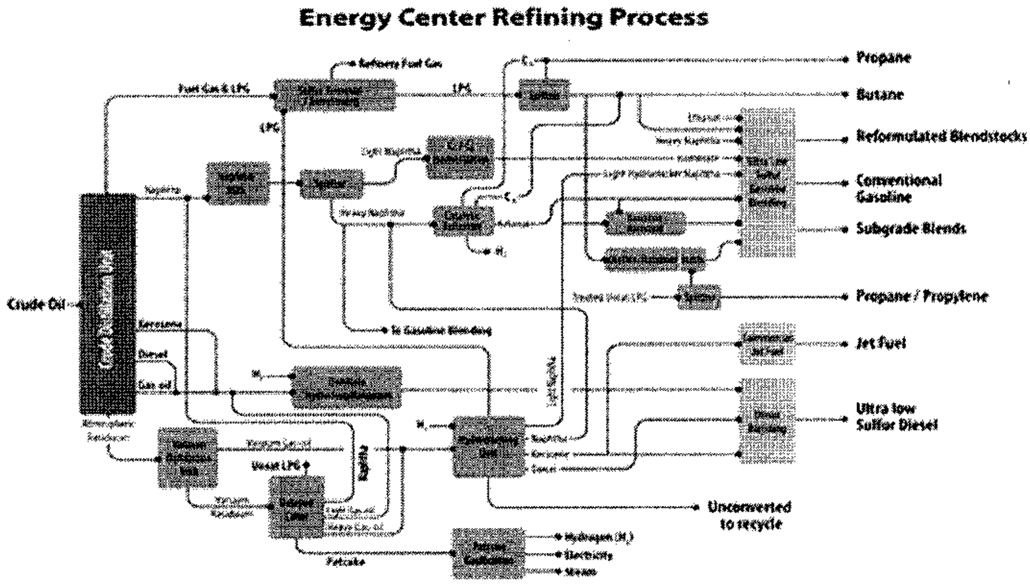


**Figure 2.1:** Refinery Process Flow (Hyperion Refining Energy Center, 2007)

The general description of refinery process flow from crude oil to product is as below:

### 2.7.1 Crude Oil

Petroleum products are made from Crude Oil. There are many types of crude oil which come from many different sources around the world. Selection of the right crude oil is a key part of the refining process. The decision as to what crude oil, or combination of crude oil, to process depends on many factors including; quality, availability, volume, and price.

### 2.7.2 Distillation

The first stage of crude processing is known as distillation, or fractionation, in a Distillation Column. The crude oil is distilled into fractions according to boiling point to yield light-end hydrocarbons ($C_1$-$C_4$), light naphtha, heavy naphtha, kerosene, diesel and atmospheric residual. Some of these broad cuts can be marketed directly while others require further processing in downstream units. Increased efficiency and reduced costs are achieved if the crude oil is fractionated at essentially atmospheric pressure followed by residue or bottoms fractionation using vacuum distillation.

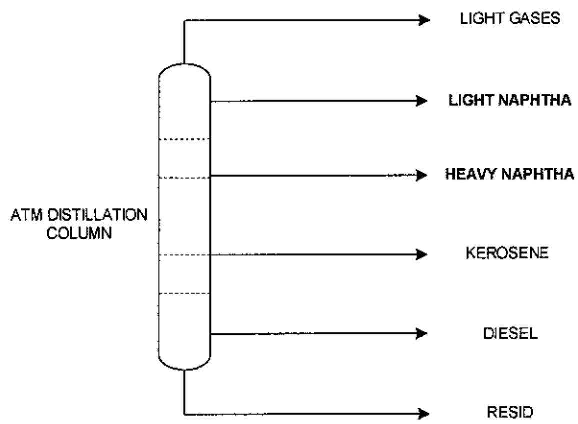


**Figure 2.2** Fractions from crude distillation

Naphtha is a complex mixture of paraffins, naphthenes and aromatics in the range of five-to-twelve carbon molecules ($C_5$ to $C_{12}$). Straight-run naphtha is obtained directly from the atmospheric distillation unit (ADU). Light naphtha is the fraction boiling from 30°C to 90°C, and contains $C_5$ and $C_6$ hydrocarbons. Heavy naphtha is the fraction boiling from 90°C to 200°C and contains $C_7$ to $C_9$ hydrocarbons, which is the favored feedstock to the catalytic reformer. Naphtha can also be sourced from the processing of heavier crude fractions in the visbreaker, catalytic cracker, hydrocracker and coker, where case olefinic hydrocarbons is present (Prestvik et al., 2004).

### 2.7.3 Hydroprocessing

Hydrotreatment is the conventional means for removing sulfur from petroleum fractions. This process is important to avoid poisoning of the reformer catalyst and to meet environmental legislations on combustion gas emissions. The feedstock is passed together with hydrogen-rich gas (usually above 75 percent hydrogen by mass), over a fixed-bed of catalyst under conditions that depend mainly on the feedstock properties and desired product specifications. Hydrodesulfurization consumes hydrogen and generates hydrogen sulfide according to the general reaction:

$$R{-}S{-}R + 2H_2 \rightarrow 2R{-}H + H_2S \qquad (14)$$

where R represents an alkyl group and S represents a sulfur atom. The severity of a hydrotreater depends on the amount and types of sulfur compounds in the naphtha feed, which in turn are determined by the crude source. Characterization of sulfur compounds in naphtha is particularly difficult due to extremely low concentrations. The sulfur composition in a blend (60 percent straight run and 40 percent hydrocracked naphtha) is almost the same as straight-run naphtha since sulfur contribution from hydrocracked naphtha is negligible. (Ali, 2004, p.114).

Catalytic naphtha hydrotreatment can simultaneously accomplish desulfurization, denitrogenation and olefin saturation. Lower-boiling compounds are desulfurized more easily than the high-boiling ones. Reactivity decreases with increasing molecular size. Products from the naphtha hydrotreater are generally acid gas, hydrogen-rich gas, LPG-rich gas and desulfurized naphtha. Table 2.1 presents the yields in terms of weight fraction for feed and products of the naphtha HDT (Parkash, 2003, p.37). The desulfurized naphtha from the hydrotreater can also be categorized as light and heavy.

**Table 2.3** Naphtha hydrotreater unit yields (weight fraction)

| Component | | Yield |
|---|---|---|
| Feed | Naphtha<br>$H_2$ | 1.0000<br>0.0080 |
| | Total | 1.0080 |
| Products | Acid gas<br>$H_2$-rich gas<br>LPG-rich gas<br>Desulfurized naphtha | 0.0012<br>0.0110<br>0.0058<br>0.9900 |
| | Total | 1.0080 |

The hydrodesulfurization of organosulfur compounds is exothermic. The amount of heat released increases with the number of moles of hydrogen consumed. This heat of reaction can increase the reactor temperature by 10-80°C at typical operating conditions, depending on the feedstock (Ali, 2004, p.119). The conditions typically used to hydrotreat straight-run stocks are mild, whereas treating cracked feeds (or blends of cracked and straight-run feeds) requires more severe conditions. Appendices A.4 presents typical ranges of conditions for three hydrotreating cases: Straight-run, Cracked or Blends of HSR and Cracked, and Synthetic or Blends of HSR and Synthetic (Ali, 2004, p.135). The principal operating variables are temperature, hydrogen partial pressure and space velocity. In general, an increase in temperature and hydrogen partial pressure increases the reaction rates of sulfur and nitrogen removal, while an increase in space velocity has the reverse effect.

### 2.7.4 Sulfur Recovery

The hydrogen sulfide generated in the hydrotreater is sent to the sulfur recovery unit before being burned as refinery gas. The conversion of hydrogen sulfide to elemental sulfur is necessary to minimize atmospheric pollution by sulfur dioxide. This is in line with environmental regulations which mandate the recovery of 99% or more of the sulfur in the refinery gas (Gary and Handwerk, 2001, p.273). The sulfur recovery unit operates based on the Claus process which proceeds as follow:

Burner : $2H_2S + 3O_2 \rightarrow 2H_2O + 2SO_2$ (15)

Reactor : $2H_2S + SO_2 \rightarrow 2H_2O + 3S$ (16)

One-third of the $H_2S$ is converted to $SO_2$ by combustion, which is then combined with the remaining two-thirds and passed over a catalyst where molten sulfur forms and is separated from the gas stream. This sulfur is sold to generate additional revenue. The gas stream is cooled by steam generation and passed over another catalyst bed. This cycle is repeated for as many as 4 catalyst beds in some instances. The gas stream leaving the sulfur recovery unit still contains $H_2S$ and/or $SO_2$ which requires further treatment (Maples, 2000, p. 351). Table 2.4 shows the product yields from a sulfur recovery unit (Parkash, 2003, p. 225).

**Table 2.4** Sulfur recovery unit yields (weight fraction)

| Component | | Yield |
|---|---|---|
| Feed | $H_2S$ gas | 1.0000 |
| | Total | 1.0000 |
| Products | Sulfur | 0.8478 |
| | Loss | 0.1522 |
| | Total | 1.0000 |

### 2.7.5 Reforming

The demand of today's automobiles for high-octane gasoline has stimulated the use of catalytic reforming to produce high-octane reformate from desulfurized naphtha without changing the boiling point range, as well as to provide hydrogen required for hydrotreating. The typical feedstocks to reformers are heavy straight-run naphthas and heavy hydrocracker naphthas. These are composed of four major hydrocarbon groups: paraffins, olefins, naphthenes and aromatics (PONA). The main function of a reformer is to convert paraffins and naphthenes into aromatics, subsequently producing high-octane reformate. Typical reformer feedstocks and products have the following PONA analyses (Gary and Handwerk, 2001, p.189):

**Table 2.5** Typical reformer feedstocks and products

| Component | Feed (vol%) | Product (vol%) |
|---|---|---|
| Paraffins | 30-70 | 30-50 |
| Olefins | 0-2 | 0-2 |
| Naphthenes | 20-60 | 0-3 |
| Aromatics | 7-20 | 45-60 |

The paraffins and naphthenes undergo two types of reactions in being converted to higher octane components: cyclization and isomerization. The ease and probability of either of these occurring increases with the number of carbon atoms in the molecules and it is for this reason that only heavy straight-run naphtha is used for reformer feed. Light straight-run naphtha is largely composed of lower-molecular weight paraffins that tend to crack to butane and lighter fractions and it is not economical to process this stream in a catalytic reformer. Hydrocarbons boiling above 204°C are easily hydrocracked and cause an excessive carbon laydown on the catalyst.

Desirable reactions in a reformer will lead to the formation of aromatics and iso-paraffins following these principle reactions (Maples, 2000, p.264):

1. Isomerization of n-paraffins to iso-paraffins
2. Dehydrocyclization of paraffins to aromatics
3. Dehydrogenation of naphthenes to aromatics
4. Olefins are saturated to form paraffins which then react as in (1) and (2).
5. Aromatics are essentially left unchanged.

Undesirable reactions are dealkylation of side chains of naphthenes and aromatics as well as cracking of paraffins and naphthenes. Table 2.3 shows reformer yields for 3 values of RONs (Parkash, 2003, p.116).

**Table 2.6** Catalytic reformer product yields (weight fraction)

| Component | | RON | | |
|---|---|---|---|---|
| | | **96** | **100** | **102** |
| Feed | Heavy naphtha | 1.0000 | 1.0000 | 1.0000 |
| | Total | 1.0000 | 1.0000 | 1.0000 |
| Products | $H_2$ | 0.0193 | 0.0310 | 0.0320 |
| | $C_1$ | 0.0085 | 0.0120 | 0.0140 |
| | $C_2$ | 0.0138 | 0.0200 | 0.0230 |
| | $C_3$ | 0.0269 | 0.0290 | 0.0330 |
| | $iC_4$ | 0.018 | 0.0170 | 0.0190 |
| | $nC_4$ | 0.0228 | 0.0230 | 0.0260 |
| | $iC_5$ | 0.0276 | - | - |
| | $nC_5$ | 0.0184 | - | - |
| | $C_5+$ | - | 0.8680 | 0.8530 |
| | $C_6+$ | 0.8447 | - | - |
| | Total | 1.0000 | 1.0000 | 1.0000 |

### 2.7.6 Isomerization

The octane numbers of light straight-run naphtha can be improved by isomerization to convert normal paraffins of $C_5$ and $C_6$ to their isomers. This results in a significant octane increase because n-pentane has a RON of 61.7 whereas the RON of iso-pentane is 92.3 (Gary and Handwerk, 2001, p.204). Equilibrium conversion to isomers is enhanced at lower temperatures; hence a reactor temperature of 98 to 205°C is desirable. At these low temperatures a very active catalyst is necessary to provide a reasonable reaction rate. Catalysts used for isomerization contain platinum on various bases. Small amounts of organic chlorides are injected continuously to maintain high catalyst activities. This leads to the formation of hydrogen chloride in the reactor which necessitates the feed to be free of water and other oxygen sources so that catalyst deactivation and potential corrosion problems can be avoided. An atmosphere of hydrogen is used to minimize carbon deposits on the catalyst but hydrogen consumption is negligible (Gary and Handwerk, 2001, p.205). Slight hydrocracking occurs during isomerization, resulting in loss of gasoline and production of light gases. Light straight-run naphtha is also sold as petrochemical feedstock besides being sent to the isomerization unit. Table 2.5 shows isomerization yields (Parkash, 2003, p.144).

**Table 2.7** Isomerization unit yields (weight fraction)

| Component | | Yield |
|---|---|---|
| Feed | Light naphtha feed<br>Hydrogen | 1.0000<br>0.0040 |
| | Total | 1.0040 |
| Products | Isomerate<br>Gases | 0.9940<br>0.0100 |
| | Total | 1.0040 |

### 2.7.7 Blending

The final stage of the refining process is blending. This is a crucial step where the various hydrocarbon components manufactured in the refinery are mixed together to make the final products sold by the refinery. The final blend recipes will depend on the quality of the available components and on the customer's requirements, called specifications. All blended products are tested before they are sold to ensure that they meet the customer's specifications.

# CHAPTER 3
# METHODOLOGY

## 3.1 OPTIMIZATION MODEL FORMULATION

### 3.1 1 Superstructure representation

Superstructure representation for the naphtha produced from the atmospheric distillation unit (ADU) with shows the optimized refinery topology is presented in this chapter. Figure 3.1 depicts the state–task network (STN) superstructure representation while Table 3.1 shows the Legend for modified state–task network (STN) superstructure representation in Figure 3.1

In developing the superstructure representation, integer binary 0–1 variables are employed as structural variables to represent discrete decisions involved in the selection of the alternative:

1. process units or tasks, as represented by binary variable $y_i$, and
2. material streams or states, as represented by binary variable $z_i$.

Temperatures of the streams and the process units as well as the product yields are assumed to be fixed values, that is, these values are considered to be known *a priori* and are constants.

### 3.1.2 Short descriptions of each process units used in the superstructure

1 *Atmospheric Distillation Unit (ADU)* - Atmospheric Distillation Unit perform the initial separation of crude oil into raw products, namely Gas, Naphtha, Kerosene, Diesel and Residue (Atmospheric Bottoms).

2 *Naphtha Hydrotreater (HDT)* - Naphtha Hydrotreater unit uses hydrogen to desulfurize naphtha from atmospheric distillation. The naphtha must be hydrotreated before sending to a Catalytic Reformer unit.

3 *Catalytic Reformer (REF)* - Catalytic Reformer unit is used to convert the naphtha-boiling range molecules into higher octane reformate (reformer product). The reformate has higher content of aromatics and cyclic

hydrocarbons). An important byproduct of a reformer is hydrogen released during the catalyst reaction. The hydrogen is used either in the hydrotreaters or the hydrocracker.

4 *Fluid Catalytic Cracker (FCC)* - Fluid Catalytic Cracker unit upgrades heavier fractions into lighter, more valuable products.

5 *Hydrocracker (HCR)* - Hydrocracker unit uses hydrogen to upgrade heavier fractions into lighter, more valuable products.

6 *Visbreaking unit (VIS)* - Visbreaking unit upgrades heavy residual oils by thermally cracking them into lighter, more valuable reduced viscosity products.

7 *Coking unit (COK)* - Coking units (delayed coking, fluid coker, and flexicoker) process very heavy residual oils into gasoline and diesel fuel, leaving petroleum coke as a residual product.

8 *Isomerization unit (ISO)* - Isomerization unit converts linear molecules to higher-octane branched molecules for blending into gasoline or feed to alkylation units.

### 3.1.3 Refinery Process Flow of Processing Route Alternatives for Naphtha Produced from Atmospheric Distillation Unit

The first processing step in refining is crude distillation, wherein crude oil (CR) is distilled into oil fractions with respect to its boiling points. Naphtha constitutes the lighter fractions that are obtained from this preliminary process. Depending on the distillation column design as well as the refinery economics, the atmospheric distillation unit (ADU) can produce:

1. light straight run naphtha (LSRN 1) *and* heavy straight run naphtha (HSRN 1), *or*
2. an undifferentiated class of naphtha (NAP 1).

for which, 0–1 structural variables denoted as $z_i$ are used to represent these possible states of the naphtha streams produced from the ADU. In the first case, LSRN 1 is sent to a mixer (MIX 3) together with purchased naphtha (PCHN 2) and light straight run naphtha (LSRN 2) from the hydrotreater (HDT 1). The output from MIX 3, which is LSRN 4, has two processing routes:

1. to be used as a feedstock for the isomerization unit (ISO) *and*
2. to be sold as a final product.

In the same manner, 0–1 structural variables denoted as $y_i$ are used to represent the possible tasks corresponding to the process units for the further conversion and treatment of the naphtha streams.

Isomerization yields isomerate (ISO), which is one of the blending components for gasoline (GSLN). Meanwhile, HSRN 1 is mixed with naphtha from the cracking of heavier fractions (to be elaborated shortly) in MIX 1 before being sent to HDT 1 to be desulfurized.

HDT 1 yields hydrogen sulfide gas (H2S 1), liquefied petroleum gas (LPG 1), desulfurized naphtha and fuel gas (FG 1). H2S 1 is sent to the sulfur recovery unit (SRU) where sulfur (S) is extracted and finally sold. All LPG (LPG 1, 2 and 3) is sent to MIX 6 and subsequently to the LPG recovery unit (LPG) from which treated LPG (LPG 5) is sold. Similar to the output from the distillation unit (ADU), the desulfurized naphtha from HDT 1 can be classified as light (LSRN 2) *and* heavy (HSRN 3) *or* undifferentiated (NAP 4). LSRN 2 is mixed with LSRN 1 and PCHN 2 in MIX 3, as stated above.

On the other hand, HSRN 3 is sent to a mixer (MIX 4), possibly with purchased naphtha (PCHN 3_1) *and/or* naphtha from the hydrocracker (HCR). The output of MIX 4 (HSRN 5) is the feedstock for the reformer (REF). FG 1 goes to the fuel gas header (FGH) which supplies fuel gas (FG 5) to the entire refinery. In the case that undifferentiated naphtha (NAP 4) is produced from HDT 1, it will also be mixed with purchased naphtha (PCHN 3_2) *and/or* naphtha from the hydrocracker (HCR) in MIX 5, whose output (NAP 5) is sent to the reformer. The products from the reformer are hydrogen gas (H2) fuel gas (FG 3), liquefied petroleum gas (LPG 2) and reformate (REF). H2 is a feed to the hydrotreater while reformate is used as a gasoline blending component. FG 3 is sent to the FGH.

In the second case (undifferentiated naphtha (NAP 1) is produced from the distillation unit), the processing route is similar to the first case in that NAP 1 will be mixed with naphtha from cracking processes in MIX 2 before being hydrotreated in HDT 2. The products from HDT 2 are H2S2, LPG 3, desulfurized naphtha and FG 2. Each product has the exact same route as the products from HDT 1.

Other than distillation, naphtha is also produced from the cracking of distillation bottoms in the visbreaker (VIS), catalytic cracker (FCC), hydrocracker (HCR), or coker (COK). The visbreaker has the lowest severity while the coker has the highest. Few assumptions were made in developing the superstructure for this section:

1. The intermediate products from the visbreaker (VIS), delayed coker (COK), fluidized catalytic cracker (FCC), and hydrocracker (HCR) are assumed to be heavy naphtha (that is, heavier fractions of naphtha).
2. It is assumed that the API for medium and heavy crudes is $\leq 33°$ whereas for light crude, the API is $> 33°$ (Trambouze et. al, 2001).

The processing of medium and heavy crude requires more severe processes; hence, the COK, FCC and HCR will be enforced to exist. On the other hand, the processing of light crude will require less severe processes, hence, only the VIS and FCC will be enforced to exist.

**Table 3.1:** Legend for the STN superstructure representation in Figure 3.1

| | | | |
|---|---|---|---|
| CR | Crude oil | HDT | Hydrotreater |
| ADU | Atmospheric distillation unit | LPG | Liquefied petroleum gas |
| LSRN | Light straight run naphtha | H2 | Hydrogen |
| HSRN | Heavy straight run naphtha | ISO | Isomerization unit |
| NAP | Naphtha | SRU | Sulfur recovery unit |
| MIX | Mixer | REF | Reformer |
| SPLT | Splitter | S | Sulfur |
| VIS | Visbreaker | FG | Fuel gas |
| COK | Coker | BLND | Blending |
| FCC | Fluidized catalytic cracker | FGH | Fuel gas header |
| HCR | Hydrocracker | GSLN | Gasoline |
| PCHN | Purchased naphtha | TG | Tail gas |

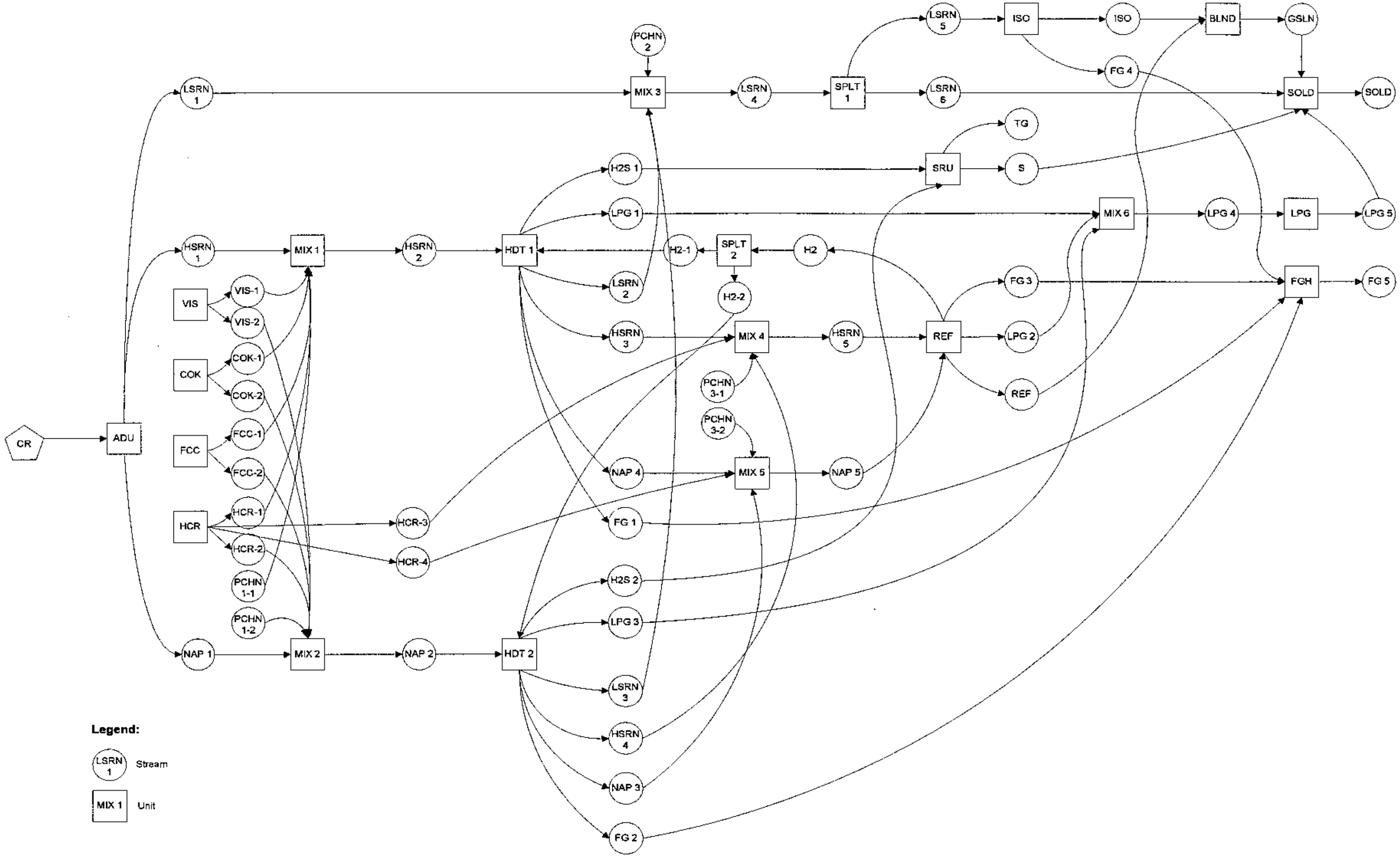


**Figure 3.1:** State–task network (STN) superstructure representation for the naphtha produced from the ADU.

### 3.1.4 Material balances for the naphtha network structure

Material balances on process units for naphtha network structure are shown in two forms:

1. the overall input–output mass flow rates, given by the following general relation:

input = output

2. the component mass balances, which incorporate the associated product yields.

**Table 3.2:** Material balances in terms of mass flow rates around the process units

| | |
|---|---|
| ADU | $0.4176 f_{CR} = f_{NAP1} + f_{LSRN1} + f_{HSRN1}$ |
| HDT 1 | $1.9821(f_{HSRN2} + f_{H2\_1}) = f_{FG1} + f_{H2S1} + f_{LPG1} + f_{LSRN2} + f_{HSRN3} + f_{NAP4}$ |
| HDT 2 | $1.9821(f_{NAP2} + f_{H2\_2}) = f_{FG2} + f_{H2S2} + f_{LPG3} + f_{LSRN3} + f_{HSRN4} + f_{NAP3}$ |
| ISO | $f_{LSRN5} = f_{ISO} + f_{FG4}$ |
| SRU | $f_{H2S1} + f_{H2S2} = f_{S} + f_{TG}$ |
| REF | $f_{HSRN5} + f_{NAP5} = f_{H2} + f_{FG3} + f_{LPG2} + f_{REF}$ |
| SOLD | $f_{LSRN6} + f_{S} + f_{GSLN} + f_{LPG5} = f_{SOLD}$ |
| BLND | $f_{ISO} + f_{REF} = f_{GSLN}$ |
| LPG | $f_{LPG4} = f_{LPG5}$ |
| FGH | $f_{FG1} + f_{FG2} + f_{FG3} + f_{FG4} = f_{FG5}$ |
| SPLT 1 | $f_{LSRN4} = f_{LSRN5} + f_{LSRN6}$ |
| SPLT 2 | $f_{H2} = f_{H2\_1} + f_{H2\_2}$ |
| MIX 1 | $f_{HSRN1} + f_{VIS\_1} + f_{COK\_1} + f_{FCC\_1} + f_{HCR\_1} + f_{PCHN1\_1} = f_{HSRN2}$ |
| MIX 2 | $f_{NAP1} + f_{VIS\_2} + f_{COK\_2} + f_{FCC\_2} + f_{HCR\_2} + f_{PCHN1\_2} = f_{NAP2}$ |
| MIX 3 | $f_{LSRN1} + f_{LSRN2} + f_{LSRN3} + f_{PCHN2} = f_{LSRN4}$ |
| MIX 4 | $f_{HSRN3} + f_{HSRN4} + f_{PCHN3\_1} + f_{HCR\_3} = f_{HSRN5}$ |
| MIX 5 | $f_{NAP3} + f_{NAP4} + f_{PCHN3\_2} + f_{HCR\_4} = f_{NAP5}$ |
| MIX 6 | $f_{LPG1} + f_{LPG2} + f_{LPG3} = f_{LPG4}$ |

**Table 3.3:** Material balances in terms of component mass flowrates around process units (Maples, 2000, p. 96; Parkash, 2003, pp. 37, 116, 225, 144)

| | |
|---|---|
| ADU | $(0.0555)\ f_{CR} = f_{LSRN\ 1}$<br>$(0.1533)\ f_{CR} = f_{HSRN\ 1}$<br>$(0.2088)\ f_{CR} = f_{NAP\ 1}$ |
| HDT 1 | $0.0109\ (f_{H2\_1} + f_{HSRN\ 2}) = f_{FG\ 1}$<br>$0.0012\ (f_{H2\_1} + f_{HSRN\ 2}) = f_{H2S\ 1}$<br>$0.0058\ (f_{H2\_1} + f_{HSRN\ 2}) = f_{LPG\ 1}$<br>$0.2610\ (f_{H2\_1} + f_{HSRN\ 2}) = f_{LSRN\ 2}$<br>$0.7211\ (f_{H2\_1} + f_{HSRN\ 2}) = f_{HSRN\ 3}$<br>$0.9821\ (f_{H2\_1} + f_{HSRN\ 2}) = f_{NAP\ 4}$ |
| HDT 2 | $0.0109\ (f_{H2\_2} + f_{NAP\ 2}) = f_{FG\ 2}$<br>$0.0012\ (f_{H2\_2} + f_{NAP\ 2}) = f_{H2S\ 2}$<br>$0.0058\ (f_{H2\_2} + f_{NAP\ 2}) = f_{LPG\ 3}$<br>$0.2610\ (f_{H2\_2} + f_{NAP\ 2}) = f_{LSRN\ 3}$<br>$0.7211\ (f_{H2\_2} + f_{NAP\ 2}) = f_{HSRN\ 4}$<br>$0.9821\ (f_{H2\_2} + f_{NAP\ 2}) = f_{NAP\ 3}$ |
| ISO | $(0.9900)\ f_{LSRN\ 5} = f_{ISO}$<br>$(0.0100)\ f_{LSRN\ 5} = f_{FG\ 4}$ |
| SRU | $0.8478\ (f_{H2S\ 1} + f_{H2S\ 2}) = f_{S}$<br>$0.1522\ (f_{H2S\ 1} + f_{H2S\ 2}) = f_{TG}$ |
| REF (based on RON = 102) | $0.0320\ (f_{HSRN\ 5} + f_{NAP\ 5}) = f_{H2}$<br>$0.0370\ (f_{HSRN\ 5} + f_{NAP\ 5}) = f_{FG\ 3}$<br>$0.0780\ (f_{HSRN\ 5} + f_{NAP\ 5}) = f_{LPG\ 2}$<br>$0.8530\ (f_{HSRN\ 5} + f_{NAP\ 5}) = f_{REF}$ |
| SPLT 1 | $(0.9000)\ f_{LSRN\ 4} = f_{LSRN\ 5}$<br>$(0.1000)\ f_{LSRN\ 4} = f_{LSRN\ 6}$ |

### 3.1.5 Big-*M* Logical constraint relating continuous and binary variables

To ensure that the non-existence of a process unit results in the corresponding input flowrates to the unit assuming the value of zero, we consider the formulation of big-*M* logical constraints to impose the relations between:

1. the continuous variables representing the flow rates of the streams and
2. the discrete binary variables representing the existence of streams and process units.

The general formulation of the big-*M* logical constraints are given by

$$f_i \leq M_i y_i$$

where $f_i$ = flow rate of output stream for process unit $i$ in kg/day,
$M_i$ = maximum capacity of process unit $i$ in kg/d,
$y_i$ = existence of process unit $i$.

The big-*M* logical constraints are also sometimes termed as switching constraints in the literature (Rardin, 1979). As mentioned, the main function of the switching constraints is to enforce the condition that no output flow exists if the associated unit does not exist. By extension, these constraints can be written as $f_i \leq M_i z_i$ to relate the stream flowrate to the binary variable $z_i$ denoting the existence of the stream itself (instead of the unit from where it is produced). In our proposed approach, this is written for each stream with the big-*M* constant $M_i$ taken to be an arbitrarily large number, $1 \times 10^8$. Examples are:

$$\begin{aligned}
&f_{LSRN1} \leq 100000000\ z_{LSRN1} && f_{HSRN2} \leq 100000000\ z_{HSRN2} \\
&f_{HSRN1} \leq 100000000\ z_{HSRN1} && f_{NAP2} \leq 100000000\ z_{NAP2} \qquad (6) \\
&f_{NAP1} \leq 100000000\ z_{NAP1} && f_{PCHN1} \leq 100000000\ z_{PCHN1}
\end{aligned}$$

### 3.1.6 Logical constraint on design and structural specifications for selection of process units

Logical constraints are employed in this work according to the approach established by Raman and Grossmann (1991, 1992, 1993a, 1993b, 1994) to assist in obtaining the optimal topology of the refinery network structure. The roles of the logical constraints in determining the optimal refinery topology are:

1. to enforce the design specifications on the selection of the process units, and the material streams that are linking the units. These specifications are primarily based on engineering knowledge and past design experience, which mainly describe the inherent characteristics of the units;
2. to enforce the structural specifications that stipulate the interconnectivity relationships among the nodes in the network that are made up of the states (streams) and the tasks (units). These relationships describe the sequence in which the streams are linking the process units.

The logical Boolean variables are employed with values of either true (1) or false (0):

1. Boolean variables $Y_i$ to denote the existence of a task $i$ in the superstructure, which involves the process units, mixers and splitters, and
2. Boolean variables $Z_i$ to denote the existence of a state $j$ (sources and sinks).

**Table 3.4:** Logical constraints on design specifications for processing alternatives of naphtha produced from the atmospheric distillation unit

<table>
<tr><th colspan="2">Logic proposition/Logical statement</th><th>Logical expression and clauses</th><th>Algebraic integer linear inequality constraint</th><th>Desired binary variable output</th></tr>
<tr><td>1.</td><td>ADU must exist</td><td>$Y_{ADU}$<br>$(Z_{LSRN\,1} \veebar Z_{NAP\,1}) \Leftrightarrow Y_{ADU}$</td><td>$y_{ADU} = 1$<br>$z_{LSRN\,1} + z_{NAP\,1} - y_{ADU} = 0$</td><td><table><tr><th>$z_{LSRN\,1}$</th><th>$z_{NAP\,1}$</th><th>$y_{ADU}$</th></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>2.</td><td>(HDT 1 or HDT 2) exists if and only if ADU exists.</td><td>$Y_{ADU} \Leftrightarrow (Y_{HDT\,1} \veebar Y_{HDT\,2})$</td><td>$y_{HDT\,1} + y_{HDT\,2} - y_{ADU} = 0$</td><td><table><tr><th>$y_{HDT\,1}$</th><th>$y_{HDT\,2}$</th><th>$y_{ADU}$</th></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>3.</td><td>SRU exists if and only if (HDT 1 or HDT 2) exists.</td><td>$(Z_{H2S\,1} \veebar Z_{H2S\,2}) \Leftrightarrow Y_{SRU} \Leftrightarrow Y_{ADU}$</td><td>$z_{H2S\,1} + z_{H2S\,2} - y_{SRU} = 0$<br>$y_{SRU} = y_{ADU}$</td><td><table><tr><th>$y_{HDT\,1}$</th><th>$y_{HDT\,2}$</th><th>$y_{ADU}$</th><th>$y_{SRU}$</th></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>4.</td><td>MIX 3 exists if and only if (LSRN 1 or LSRN 3) exists</td><td>$(Z_{LSRN\,1} \vee Z_{LSRN\,3}) \Leftrightarrow Y_{MIX\,3}$</td><td>$z_{LSRN\,1} + z_{LSRN\,3} = y_{MIX\,3}$</td><td><table><tr><th>$z_{LSRN\,1}$</th><th>$z_{LSRN\,3}$</th><th>$y_{MIX\,3}$</th></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
<tr><td>5.</td><td>MIX 4 exists if and only if (HSRN3 or HSRN 4) existss</td><td>$(Z_{HSRN\,3} \vee Z_{HSRN\,4}) \Leftrightarrow Y_{MIX\,4}$</td><td>$z_{HSRN\,3} + z_{HSRN\,4} - y_{MIX\,4} = 0$</td><td><table><tr><th>$z_{HSRN\,3}$</th><th>$z_{HSRN\,4}$</th><th>$y_{MIX\,4}$</th></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
<tr><td>6.</td><td>MIX 5 exists if and only if (NAP 3 or NAP 4) exists</td><td>$(Z_{NAP\,3} \vee Z_{NAP\,4}) \Leftrightarrow Y_{MIX\,5}$</td><td>$z_{NAP\,3} + z_{NAP\,4} - y_{MIX\,5} = 0$</td><td><table><tr><th>$z_{NAP\,3}$</th><th>$z_{NAP\,4}$</th><th>$y_{MIX\,5}$</th></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td>5) or MIX 5 exist if and only if (HDT1 or HDT 2) exists</td><td></td><td>$y_{MIX3}+y_{MIX4}+y_{MIX5} \ge y_{HDT1}$<br>$y_{MIX3}+2*y_{MIX5} \ge y_{HDT1}$<br>$y_{MIX4}+2*y_{MIX5} \ge y_{HDT1}$<br>$2*y_{MIX3}+y_{MIX5} \ge y_{HDT2}$<br>$y_{MIX3}+y_{MIX4}+y_{MIX5} \ge y_{HDT2}$<br>$y_{MIX3}+2*y_{MIX5} \ge y_{HDT2}$<br>$y_{MIX4}+2*y_{MIX5} \ge y_{HDT2}$</td><td><table>
<tr><th>$y_{HDT1}$</th><th>$y_{HDT2}$</th><th>$y_{MIX3}$</th><th>$y_{MIX4}$</th><th>$y_{MIX5}$</th></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>8.</td><td>REF exists if and only if (HDT 1 or HDT 2) exists</td><td>$(Z_{HSRN5} \underline{\vee} Z_{NAP5}) \Leftrightarrow Y_{REF} \Leftrightarrow Y_{ADU}$</td><td>$y_{REF} = y_{ADU}$<br>$z_{HSRN5} + z_{NAP5} - y_{REF} = 0$</td><td><table>
<tr><th>$z_{HSRN5}$</th><th>$z_{NAP5}$</th><th>$y_{REF}$</th><th>$y_{ADU}$</th></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>9.</td><td>LPG exists if and only if (HDT 1 or HDT 2) exists</td><td>$(Y_{HDT1} \vee Y_{HDT2}) \Leftrightarrow Y_{LPG} \Leftrightarrow Y_{ADU}$<br>$(Z_{LPG1} \vee Z_{LPG2} \vee Z_{LPG3}) \Leftrightarrow Y_{LPG}$</td><td>$y_{LPG} = y_{ADU} = 1$<br>$y_{HDT1} + y_{HDT2} - y_{LPG} = 0$<br>$y_{HDT1} + y_{HDT2} = 1$<br>$z_{LPG1} + z_{LPG2} + z_{LPG3} - y_{LPG} = 1$<br>$y_{HDT1} + y_{HDT2} = z_{LPG1} + z_{LPG2} + z_{LPG3} - y_{LPG}$</td><td><table>
<tr><th>$z_{LPG1}$</th><th>$z_{LPG2}$</th><th>$z_{LPG3}$</th><th>$y_{LPG}$</th><th>$y_{HDT1}$</th><th>$y_{HDT2}$</th></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table></td></tr>
<tr><td>10.</td><td>FGH exists if and only if (HDT 1 or HDT 2 or ISO) exists</td><td>$(Z_{FG1} \vee Z_{FG2} \vee Z_{FG3} \vee Z_{FG4}) \Leftrightarrow Y_{FGH}$</td><td>$y_{FGH} \ge z_{FG1}$<br>$y_{FGH} \ge z_{FG2}$<br>$y_{FGH} \ge z_{FG3}$<br>$y_{FGH} \ge z_{FG4}$<br>$z_{FG1} + z_{FG2} + z_{FG3} + z_{FG4} \ge y_{FGH}$</td><td><table>
<tr><th>$z_{FG1}$</th><th>$z_{FG2}$</th><th>$z_{FG3}$</th><th>$y_{FG4}$</th><th>$y_{FGH}$</th></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>11.</td><td>ISO exists if and only if HDT 1 exists</td><td>$Y_{ISO} \Leftrightarrow Y_{HDT1}$</td><td>$y_{ISO} = y_{HDT1}$</td><td><table>
<tr><th>$y_{ISO}$</th><th>$y_{HDT1}$</th></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td></tr>
</table></td></tr>
<tr><td>12.</td><td>BLND exists if and only if ( $Z_{ISO}$ or/and $Z_{REF}$ ) exist</td><td>$(Z_{ISO} \vee Z_{REF}) \Leftrightarrow Y_{BLND}$</td><td>$y_{BLND} \ge z_{ISO}$<br>$y_{BLND} \ge z_{REF}$<br>$z_{ISO} + z_{REF} \ge y_{BLND}$</td><td><table>
<tr><th>$z_{ISO}$</th><th>$z_{REF}$</th><th>$y_{BLND}$</th></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
</table></td></tr>
<tr><td>13.</td><td>SOLD exists if and only if ADU exists</td><td>$Y_{SOLD} = Y_{ADU}$</td><td>$y_{SOLD} = y_{ADU}$</td><td><table>
<tr><th>$y_{SOLD}$</th><th>$y_{ADU}$</th></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td></tr>
</table></td></tr>
</table>

Table 3.4 shows the logical statement or logic propositions, which can be equivalently expressed in the form of logical expression and clauses. The logical expression and clauses are then transformed into algebraic integer linear inequality constraints by applying the systematic procedure proposed by Raman and Grossmann (1991), as discussed in Chapter 2.

The following are two examples on the transformation of a logic proposition into its corresponding algebraic integer constraint:

- example 1: for the logical statement "EOR" for ADU (item no. 1 in Table 4.2)

$$Z_{\text{LSRN 1}} \veebar Z_{\text{NAP 1}} \Leftrightarrow Y_{\text{ADU}}$$
$$z_{\text{LSRN 1}} + z_{\text{NAP 1}} = y_{\text{ADU}}$$

- example 2: for the logical statement "OR" for FGH (item no. 10 in Table 4.2)

$$\left(Z_{\text{FG1}} \vee Z_{\text{FG2}} \vee Z_{\text{FG3}} \vee Z_{\text{FG4}}\right) \Leftrightarrow Y_{\text{FGH}}$$
$$\left\{\left(Z_{\text{FG1}} \vee Z_{\text{FG2}} \vee Z_{\text{FG3}} \vee Z_{\text{FG4}}\right) \Rightarrow Y_{\text{FGH}}\right\} \wedge \left\{Y_{\text{FGH}} \Rightarrow \left(Z_{\text{FG1}} \vee Z_{\text{FG2}} \vee Z_{\text{FG3}} \vee Z_{\text{FG4}}\right)\right\}$$

$$\left(Z_{\text{FG1}} \vee Z_{\text{FG2}} \vee Z_{\text{FG3}} \vee Z_{\text{FG4}}\right) \Rightarrow Y_{\text{FGH}}$$
$$\neg\left(Z_{\text{FG1}} \vee Z_{\text{FG2}} \vee Z_{\text{FG3}} \vee Z_{\text{FG4}}\right) \vee Y_{\text{FGH}}$$
$$\left(\neg Z_{\text{FG1}} \wedge \neg Z_{\text{FG2}} \wedge \neg Z_{\text{FG3}} \wedge \neg Z_{\text{FG4}}\right) \vee Y_{\text{FGH}}$$
$$\left(\neg Z_{\text{FG1}} \vee Y_{\text{FGH}}\right) \wedge \left(\neg Z_{\text{FG2}} \vee Y_{\text{FGH}}\right) \wedge \left(\neg Z_{\text{FG3}} \vee Y_{\text{FGH}}\right) \wedge \left(\neg Z_{\text{FG4}} \vee Y_{\text{FGH}}\right)$$
$$1 - z_{\text{FG1}} + y_{\text{FGH}} \geq 1 \qquad y_{\text{FGH}} \geq z_{\text{FG2}} \qquad y_{\text{FGH}} \geq z_{\text{FG3}} \qquad y_{\text{FGH}} \geq z_{\text{FG4}}$$
$$y_{\text{FGH}} \geq z_{\text{FG1}}$$
$$Y_{\text{FGH}} \Rightarrow \left(Z_{\text{FG1}} \vee Z_{\text{FG2}} \vee Z_{\text{FG3}} \vee Z_{\text{FG4}}\right)$$
$$\neg Y_{\text{FGH}} \vee \left(Z_{\text{FG1}} \vee Z_{\text{FG2}} \vee Z_{\text{FG3}} \vee Z_{\text{FG4}}\right)$$
$$z_{\text{FG 1}} + z_{\text{FG 2}} + z_{\text{FG 3}} + z_{\text{FG 4}} \geq y_{\text{FGH}}$$

Therefore, the result is

$$y_{\text{FGH}} \geq z_{\text{FG 1}}$$
$$y_{\text{FGH}} \geq z_{\text{FG 2}}$$
$$y_{\text{FGH}} \geq z_{\text{FG 3}}$$
$$y_{\text{FGH}} \geq z_{\text{FG4}}$$
$$z_{\text{FG 1}} + z_{\text{FG 2}} + z_{\text{FG 3}} + z_{\text{FG 4}} \geq y_{\text{FGH}}$$

### 3.1.8 Logic Prepositions and Algebraic Integer Constraints for Structural Specifications

The structural specification is to enforce the interconnectivity among the states and the tasks in the superstructure. The detail on the generation of the logical constraints on structural specification and the transformation into algebraic integer linear inequality constraints are shown in Table 3.5.

**Table 3.5:** Logical Preposition and Algebraic Integer for Structural Specification

| 'nit | Logic Preposition for Structural Specification | Algebraic Integer Constraint |
|---|---|---|
| DU | $Y_{ADU} \Rightarrow Z_{CR}$ | $y_{ADU} \leq z_{CR}$ |
| | $Y_{ADU} \Rightarrow Z_{LSRN\,1} \vee Z_{HSRN\,1} \vee Z_{NAP\,1}$ | $y_{ADU} \leq z_{LSRN\,1} + z_{HSRN\,1} + z_{NAP\,1}$ |
| | $Z_{LSRN\,1} \Rightarrow Y_{ADU}$ | $z_{LSRN\,1} \leq y_{ADU}$ |
| | $Z_{HSRN\,1} \Rightarrow Y_{ADU}$ | $z_{HSRN\,1} \leq y_{ADU}$ |
| | $Z_{NAP\,1} \Rightarrow Y_{ADU}$ | $z_{NAP\,1} \leq y_{ADU}$ |
| )T 1 | $Y_{HDT\,1} \Rightarrow Z_{HSRN\,2} \vee Z_{H2\text{-}1}$ | $y_{HDT\,1} \leq z_{HSRN\,2} + z_{H2\text{-}1}$ |
| | $Z_{HSRN\,2} \Rightarrow Y_{HDT\,1}$ | $z_{HSRN\,2} \leq y_{HDT\,1}$ |
| | $Z_{H2\text{-}1} \Rightarrow Y_{HDT\,1}$ | $z_{H2\text{-}1} \leq y_{HDT\,1}$ |
| | $Y_{HDT\,1} \Rightarrow \begin{pmatrix} Z_{H2S\,1} \vee Z_{LPG\,1} \vee Z_{LSRN\,2} \\ \vee Z_{HSRN\,3} \vee Z_{NAP\,4} \vee Z_{FG\,1} \end{pmatrix}$ | $y_{HDT\,1} \leq z_{H2S\,1} + z_{LPG\,1} + z_{LSRN\,2} + z_{HSRN\,3} + z_{NAP\,4} + z_{FG\,1}$ |
| | | $z_{H2S\,1} \leq y_{HDT\,1}$ |
| | $Z_{H2S\,1} \Rightarrow Y_{HDT\,1}$ | $z_{LPG\,1} \leq y_{HDT\,1}$ |
| | $Z_{LPG\,1} \Rightarrow Y_{HDT\,1}$ | $z_{LSRN\,2} \leq y_{HDT\,1}$ |
| | $Z_{LSRN\,2} \Rightarrow Y_{HDT\,1}$ | $z_{HSRN\,3} \leq y_{HDT\,1}$ |
| | $Z_{HSRN\,3} \Rightarrow Y_{HDT\,1}$ | $z_{NAP\,4} \leq y_{HDT\,1}$ |
| | $Z_{NAP\,4} \Rightarrow Y_{HDT\,1}$ | $z_{FG\,1} \leq y_{HDT\,1}$ |
| | $Z_{FG\,1} \Rightarrow Y_{HDT\,1}$ | |

| | | |
|---|---|---|
| DT 2 | $Y_{HDT\,2} \Rightarrow Z_{NAP\,2} \vee Z_{H2\text{-}2}$ | $y_{HDT\,2} \leq z_{NAP\,2} + z_{H2\text{-}2}$ |
| | $Z_{NAP\,2} \Rightarrow Y_{HDT\,2}$ | $z_{NAP\,2} \leq y_{HDT\,2}$ |
| | $Z_{H2\text{-}2} \Rightarrow Y_{HDT\,2}$ | $z_{H2\text{-}2} \leq y_{HDT\,2}$ |
| | $Y_{HDT\,2} \Rightarrow \begin{pmatrix} Z_{H2S\,2} \vee Z_{LPG\,3} \vee Z_{LSRN\,3} \\ \vee Z_{HSRN\,4} \vee Z_{NAP\,3} \vee Z_{FG\,2} \end{pmatrix}$ | $y_{HDT\,2} \leq z_{H2S\,2} + z_{LPG\,3} + z_{LSRN\,3} + z_{HSRN\,4} + z_{NAP\,3} + z_{FG\,2}$ |
| | | $z_{H2S\,2} \leq y_{HDT\,2}$ |
| | $Z_{H2S\,2} \Rightarrow Y_{HDT\,2}$ | $z_{LPG\,3} \leq y_{HDT\,2}$ |
| | $Z_{LPG\,3} \Rightarrow Y_{HDT\,2}$ | $z_{LSRN\,3} \leq y_{HDT\,2}$ |
| | $Z_{LSRN\,3} \Rightarrow Y_{HDT\,2}$ | $z_{HSRN\,4} \leq y_{HDT\,2}$ |
| | $Z_{HSRN\,4} \Rightarrow Y_{HDT\,2}$ | $z_{NAP\,3} \leq y_{HDT\,2}$ |
| | $Z_{NAP\,3} \Rightarrow Y_{HDT\,2}$ | $z_{FG\,2} \leq y_{HDT\,2}$ |
| | $Z_{FG\,2} \Rightarrow Y_{HDT\,2}$ | |
| SO | $Z_{LSRN\,5} \Rightarrow Y_{ISO}$ | $z_{LSRN\,5} \leq y_{ISO}$ |
| | $Y_{ISO} \Rightarrow Z_{LSRN\,5}$ | $y_{ISO} \leq z_{LSRN\,5}$ |
| | $Y_{ISO} \Rightarrow Z_{ISO} \vee Z_{FG\,4}$ | $y_{ISO} \leq z_{ISO} + z_{FG\,4}$ |
| | $Z_{ISO} \Rightarrow Y_{ISO}$ | $z_{ISO} \leq y_{ISO}$ |
| | $Z_{FG\,4} \Rightarrow Y_{ISO}$ | $z_{FG\,4} \leq y_{ISO}$ |
| RU | $Y_{SRU} \Rightarrow Z_{H2S1} \vee Z_{H2S\,2}$ | $y_{SRU} \leq z_{H2S1} + z_{H2S\,2}$ |
| | $Z_{H2S1} \Rightarrow Y_{SRU}$ | $z_{H2S1} \leq y_{SRU}$ |
| | $Z_{H2S\,2} \Rightarrow Y_{SRU}$ | $z_{H2S\,2} \leq y_{SRU}$ |
| | $Y_{SRU} \Rightarrow Z_{S} \vee Z_{TG}$ | $y_{SRU} \leq z_{S} + z_{TG}$ |
| | $Z_{S} \Rightarrow Y_{SRU}$ | $z_{S} \leq y_{SRU}$ |
| | $Z_{TG} \Rightarrow Y_{SRU}$ | $z_{TG} \leq y_{SRU}$ |
| EF | $Y_{REF} \Rightarrow Z_{HSRN\,5} \vee Z_{NAP\,5}$ | $y_{REF} \leq z_{HSRN\,5} + z_{NAP\,5}$ |
| | $Z_{HSRN\,5} \Rightarrow Y_{REF}$ | $z_{HSRN\,5} \leq y_{REF}$ |
| | $Z_{NAP\,5} \Rightarrow Y_{REF}$ | $z_{NAP\,5} \leq y_{REF}$ |
| | $Y_{REF} \Rightarrow Z_{H2} \vee Z_{FG\,3} \vee Z_{LPG\,2} \vee Z_{REF}$ | $y_{REF} \leq z_{H2} + z_{FG\,3} + z_{LPG\,2} + z_{REF}$ |
| | $Z_{H2} \Rightarrow Y_{REF}$ | $z_{H2} \leq y_{REF}$ |
| | $Z_{FG\,3} \Rightarrow Y_{REF}$ | $z_{FG\,3} \leq y_{REF}$ |
| | $Z_{LPG\,2} \Rightarrow Y_{REF}$ | $z_{LPG\,2} \leq y_{REF}$ |
| | $Z_{REF} \Rightarrow Y_{REF}$ | $z_{REF} \leq y_{REF}$ |
| LD | $Y_{SOLD} \Rightarrow Z_{LSRN6} \vee Z_{GSLN} \vee Z_{S} \vee Z_{LPG\,5}$ | $y_{SOLD} \leq z_{LSRN6} + z_{GSLN} + z_{S} + z_{LPG\,5}$ |
| | $Z_{LSRN6} \Rightarrow Y_{SOLD}$ | $z_{LSRN6} \leq y_{SOLD}$ |
| | $Z_{GSLN} \Rightarrow Y_{SOLD}$ | $z_{GSLN} \leq y_{SOLD}$ |
| | $Z_{S} \Rightarrow Y_{SOLD}$ | $z_{S} \leq y_{SOLD}$ |
| | $Z_{LPG\,5} \Rightarrow Y_{SOLD}$ | $z_{LPG\,5} \leq y_{SOLD}$ |
| | $Y_{SOLD} \Rightarrow Z_{SOLD}$ | $y_{SOLD} \leq z_{SOLD}$ |
| | $Z_{SOLD} \Rightarrow Y_{SOLD}$ | $z_{SOLD} \leq y_{SOLD}$ |

| | | |
|---|---|---|
| LND | $Y_{BLND} \Rightarrow Z_{ISO} \vee Z_{REF}$ | $y_{BLND} \le z_{ISO} + z_{REF}$ |
| | $Z_{ISO} \Rightarrow Y_{BLND}$ | $z_{ISO} \le y_{BLND}$ |
| | $Z_{REF} \Rightarrow Y_{BLND}$ | $z_{REF} \le y_{BLND}$ |
| | $Y_{BLND} \Rightarrow Z_{GSLN}$ | $y_{BLND} \le z_{GSLN}$ |
| | $Z_{GSLN} \Rightarrow Y_{BLND}$ | $z_{GSLN} \le y_{BLND}$ |
| PG | $Z_{LPG\,4} \Rightarrow Y_{LPG}$ | $z_{LPG\,4} \le y_{LPG}$ |
| | $Y_{LPG} \Rightarrow Z_{LPG\,4}$ | $y_{LPG} \le z_{LPG\,4}$ |
| | $Y_{LPG} \Rightarrow Z_{LPG\,5}$ | $y_{LPG} \le z_{LPG\,5}$ |
| | $Z_{LPG\,5} \Rightarrow Y_{LPG}$ | $z_{LPG\,5} \le y_{LPG}$ |
| GH | $Y_{FGH} \Rightarrow Z_{FG\,1} \vee Z_{FG\,2} \vee Z_{FG\,3} \vee Z_{FG\,4}$ | $y_{FGH} \le z_{FG\,1} \vee z_{FG\,2} \vee z_{FG\,3} \vee z_{FG\,4}$ |
| | $Z_{FG\,1} \Rightarrow Y_{FGH}$ | $z_{FG\,1} \le y_{FGH}$ |
| | $Z_{FG\,2} \Rightarrow Y_{FGH}$ | $z_{FG\,2} \le y_{FGH}$ |
| | $Z_{FG\,3} \Rightarrow Y_{FGH}$ | $z_{FG\,3} \le y_{FGH}$ |
| | $Z_{FG\,4} \Rightarrow Y_{FGH}$ | $z_{FG\,4} \le y_{FGH}$ |
| | $Y_{FGH} \Rightarrow Z_{FG\,5}$ | $y_{FGH} \le z_{FG\,5}$ |
| | $Z_{FG\,5} \Rightarrow Y_{FGH}$ | $z_{FG\,5} \le y_{FGH}$ |
| PLT 1 | $Z_{LSRN\,4} \Rightarrow Y_{SPLT\,1}$ | $z_{LSRN\,4} \le y_{SPLT\,1}$ |
| | $Y_{SPLT\,1} \Rightarrow Z_{LSRN\,4}$ | $y_{SPLT\,1} \le z_{LSRN\,4}$ |
| | $Y_{SPLT\,1} \Rightarrow Z_{LSRN\,5} \vee Z_{LSRN\,6}$ | $y_{SPLT\,1} \le z_{LSRN\,5} + z_{LSRN\,6}$ |
| | $Z_{LSRN\,5} \Rightarrow Y_{SPLT\,1}$ | $z_{LSRN\,5} \le y_{SPLT\,1}$ |
| | $Z_{LSRN\,6} \Rightarrow Y_{SPLT\,1}$ | $z_{LSRN\,6} \le y_{SPLT\,1}$ |
| PLT 2 | $Z_{H2} \Rightarrow Y_{SPLT\,2}$ | $z_{H2} \le y_{SPLT\,2}$ |
| | $Y_{SPLT\,2} \Rightarrow Z_{H2}$ | $y_{SPLT\,2} \le z_{H2}$ |
| | $Y_{SPLT\,2} \Rightarrow Z_{H2\text{-}1} \vee Z_{H2\text{-}2}$ | $y_{SPLT\,2} \le z_{H2\text{-}1} + z_{H2\text{-}2}$ |
| | $Z_{H2\text{-}1} \Rightarrow Y_{SPLT\,2}$ | $z_{H2\text{-}1} \le y_{SPLT\,2}$ |
| | $Z_{H2\text{-}2} \Rightarrow Y_{SPLT\,2}$ | $z_{H2\text{-}2} \le y_{SPLT\,2}$ |
| IX 1 | $Y_{MIX\,1} \Rightarrow \begin{pmatrix} Z_{HSRN\,1} \vee Z_{VIS\text{-}1} \vee Z_{COK\text{-}1} \vee \\ Z_{FCC\text{-}1} \vee Z_{HCR\text{-}1} \vee Z_{PCHN\,1\text{-}1} \end{pmatrix}$ | $y_{MIX\,1} \Rightarrow z_{HSRN\,1} + z_{VIS\text{-}1} + z_{COK\text{-}1} + z_{FCC\text{-}1} + z_{HCR\text{-}1} + z_{PCHN\,1\text{-}}$ |
| | $Z_{HSRN\,1} \Rightarrow Y_{MIX\,1}$ | $z_{HSRN\,1} \le y_{MIX\,1}$ |
| | $Z_{VIS\text{-}1} \Rightarrow Y_{MIX\,1}$ | $z_{VIS\text{-}1} \le y_{MIX\,1}$ |
| | $Z_{COK\text{-}1} \Rightarrow Y_{MIX\,1}$ | $z_{COK\text{-}1} \le y_{MIX\,1}$ |
| | $Z_{FCC\text{-}1} \Rightarrow Y_{MIX\,1}$ | $z_{FCC\text{-}1} \le y_{MIX\,1}$ |
| | $Z_{HCR\text{-}1} \Rightarrow Y_{MIX\,1}$ | $z_{HCR\text{-}1} \le y_{MIX\,1}$ |
| | $Z_{PCHN\,1\text{-}1} \Rightarrow Y_{MIX\,1}$ | $z_{PCHN\,1\text{-}1} \le y_{MIX\,1}$ |
| | $Y_{MIX\,1} \Rightarrow Z_{HSRN\,2}$ | $y_{MIX\,1} \le z_{HSRN\,2}$ |
| | $Z_{HSRN\,2} \Rightarrow Y_{MIX\,1}$ | $z_{HSRN\,2} \le y_{MIX\,1}$ |

| | | |
|---|---|---|
| IX 2 | $Y_{\text{MIX 2}} \Rightarrow \begin{pmatrix} Z_{\text{NAP 1}} \vee Z_{\text{VIS-2}} \vee Z_{\text{COK-2}} \vee \\ Z_{\text{FCC-2}} \vee Z_{\text{HCR-2}} \vee Z_{\text{PCHN 1-2}} \end{pmatrix}$ | $y_{\text{MIX 2}} \le z_{\text{NAP 1}} + z_{\text{VIS-2}} + z_{\text{COK-2}} + z_{\text{FCC-2}} + z_{\text{HCR-2}} + z_{\text{PCHN 1-2}}$ |
| | $Z_{\text{NAP 1}} \Rightarrow Y_{\text{MIX 2}}$ | $z_{\text{NAP 1}} \le y_{\text{MIX 2}}$ |
| | $Z_{\text{VIS-2}} \Rightarrow Y_{\text{MIX 2}}$ | $z_{\text{VIS-2}} \le y_{\text{MIX 2}}$ |
| | $Z_{\text{COK-2}} \Rightarrow Y_{\text{MIX 2}}$ | $z_{\text{COK-2}} \le y_{\text{MIX 2}}$ |
| | $Z_{\text{FCC-2}} \Rightarrow Y_{\text{MIX 2}}$ | $z_{\text{FCC-2}} \le y_{\text{MIX 2}}$ |
| | $Z_{\text{HCR-2}} \Rightarrow Y_{\text{MIX 2}}$ | $z_{\text{HCR-2}} \le y_{\text{MIX 2}}$ |
| | $Z_{\text{PCHN 1-2}} \Rightarrow Y_{\text{MIX 2}}$ | $z_{\text{PCHN 1-2}} \le y_{\text{MIX 2}}$ |
| | $Y_{\text{MIX 2}} \Rightarrow Z_{\text{NAP 2}}$ | $y_{\text{MIX 2}} \le z_{\text{NAP 2}}$ |
| | $Z_{\text{NAP 2}} \Rightarrow Y_{\text{MIX 2}}$ | $z_{\text{NAP 2}} \le y_{\text{MIX 2}}$ |
| IX 3 | $Y_{\text{MIX 3}} \Rightarrow Z_{\text{LSRN 1}} \vee Z_{\text{PCHN 2}} \vee Z_{\text{LSRN 2}} \vee Z_{\text{LSRN 3}}$ | $y_{\text{MIX 3}} \le z_{\text{LSRN 1}} + z_{\text{PCHN 2}} + z_{\text{LSRN 2}} + z_{\text{LSRN 3}}$ |
| | $Z_{\text{LSRN 1}} \Rightarrow Y_{\text{MIX 3}}$ | $z_{\text{LSRN 1}} \le y_{\text{MIX 3}}$ |
| | $Z_{\text{PCHN 2}} \Rightarrow Y_{\text{MIX 3}}$ | $z_{\text{PCHN 2}} \le y_{\text{MIX 3}}$ |
| | $Z_{\text{LSRN 2}} \Rightarrow Y_{\text{MIX 3}}$ | $z_{\text{LSRN 2}} \le y_{\text{MIX 3}}$ |
| | $Z_{\text{LSRN 3}} \Rightarrow Y_{\text{MIX 3}}$ | $z_{\text{LSRN 3}} \le y_{\text{MIX 3}}$ |
| | $Y_{\text{MIX 3}} \Rightarrow Z_{\text{LSRN 4}}$ | $y_{\text{MIX 3}} \le z_{\text{LSRN 4}}$ |
| | $Z_{\text{LSRN 4}} \Rightarrow Y_{\text{MIX 3}}$ | $z_{\text{LSRN 4}} \le y_{\text{MIX 3}}$ |
| IX 4 | $Y_{\text{MIX 4}} \Rightarrow Z_{\text{HSRN 3}} \vee Z_{\text{HCR-3}} \vee Z_{\text{PCHN 3-1}} \vee Z_{\text{HSRN 4}}$ | $y_{\text{MIX 4}} \le z_{\text{HSRN 3}} + z_{\text{HCR-3}} + z_{\text{PCHN 3-1}} + z_{\text{HSRN 4}}$ |
| | $Z_{\text{HSRN 3}} \Rightarrow Y_{\text{MIX 4}}$ | $z_{\text{HSRN 3}} \le y_{\text{MIX 4}}$ |
| | $Z_{\text{HCR-3}} \Rightarrow Y_{\text{MIX 4}}$ | $z_{\text{HCR-3}} \le y_{\text{MIX 4}}$ |
| | $Z_{\text{PCHN 3-1}} \Rightarrow Y_{\text{MIX 4}}$ | $z_{\text{PCHN 3-1}} \le y_{\text{MIX 4}}$ |
| | $Z_{\text{HSRN 4}} \Rightarrow Y_{\text{MIX 4}}$ | $z_{\text{HSRN 4}} \le y_{\text{MIX 4}}$ |
| | $Y_{\text{MIX 4}} \Rightarrow Z_{\text{HSRN 5}}$ | $y_{\text{MIX 4}} \le z_{\text{HSRN 5}}$ |
| | $Z_{\text{HSRN 5}} \Rightarrow Y_{\text{MIX 4}}$ | $z_{\text{HSRN 5}} \le y_{\text{MIX 4}}$ |
| IX 5 | $Y_{\text{MIX 5}} \Rightarrow Z_{\text{NAP 3}} \vee Z_{\text{NAP 4}} \vee Z_{\text{HCR-4}} \vee Z_{\text{PCHN 3-2}}$ | $y_{\text{MIX 5}} \le z_{\text{NAP 3}} + z_{\text{NAP 4}} + z_{\text{HCR-4}} + z_{\text{PCHN 3-2}}$ |
| | $Z_{\text{NAP 3}} \Rightarrow Y_{\text{MIX 5}}$ | $z_{\text{NAP 3}} \le y_{\text{MIX 5}}$ |
| | $Z_{\text{NAP 4}} \Rightarrow Y_{\text{MIX 5}}$ | $z_{\text{NAP 4}} \le y_{\text{MIX 5}}$ |
| | $Z_{\text{HCR-4}} \Rightarrow Y_{\text{MIX 5}}$ | $z_{\text{HCR-4}} \le y_{\text{MIX 5}}$ |
| | $Z_{\text{PCHN 3-2}} \Rightarrow Y_{\text{MIX 5}}$ | $z_{\text{PCHN 3-2}} \le y_{\text{MIX 5}}$ |
| | $Y_{\text{MIX 5}} \Rightarrow Z_{\text{NAP 5}}$ | $y_{\text{MIX 5}} \le z_{\text{NAP 5}}$ |
| | $Z_{\text{NAP 5}} \Rightarrow Y_{\text{MIX 5}}$ | $z_{\text{NAP 5}} \le y_{\text{MIX 5}}$ |

| | | |
|---|---|---|
| IX 6 | $Y_{\text{MIX 6}} \Rightarrow Z_{\text{LPG 1}} \vee Z_{\text{LPG 2}} \vee Z_{\text{LPG 3}}$ | $y_{\text{MIX 6}} \leq z_{\text{LPG 1}} + z_{\text{LPG 2}} + z_{\text{LPG 3}}$ |
| | $Z_{\text{LPG 1}} \Rightarrow Y_{\text{MIX 6}}$ | $z_{\text{LPG 1}} \leq y_{\text{MIX 6}}$ |
| | $Z_{\text{LPG 2}} \Rightarrow Y_{\text{MIX 6}}$ | $z_{\text{LPG 2}} \leq y_{\text{MIX 6}}$ |
| | $Z_{\text{LPG 3}} \Rightarrow Y_{\text{MIX 6}}$ | $z_{\text{LPG 3}} \leq y_{\text{MIX 6}}$ |
| | $Y_{\text{MIX 6}} \Rightarrow Z_{\text{LPG 4}}$ | $y_{\text{MIX 6}} \leq z_{\text{LPG 4}}$ |
| | $Z_{\text{LPG 4}} \Rightarrow Y_{\text{MIX 6}}$ | $z_{\text{LPG 4}} \leq y_{\text{MIX 6}}$ |
| /IS | $Y_{\text{VIS}} \Rightarrow Z_{\text{VIS-1}} \vee Z_{\text{VIS-2}}$ | $y_{\text{VIS}} \leq z_{\text{VIS-1}} \vee z_{\text{VIS-2}}$ |
| | $Z_{\text{VIS-1}} \Rightarrow Y_{\text{VIS}}$ | $z_{\text{VIS-1}} \leq y_{\text{VIS}}$ |
| | $Z_{\text{VIS-2}} \Rightarrow Y_{\text{VIS}}$ | $z_{\text{VIS-2}} \leq y_{\text{VIS}}$ |
| 'OK | $Y_{\text{COK}} \Rightarrow Z_{\text{COK-1}} \vee Z_{\text{COK-2}}$ | $y_{\text{COK}} \leq z_{\text{COK-1}} + z_{\text{COK-2}}$ |
| | $Z_{\text{COK-1}} \Rightarrow Y_{\text{COK}}$ | $z_{\text{COK-1}} \leq y_{\text{COK}}$ |
| | $Z_{\text{COK-2}} \Rightarrow Y_{\text{COK}}$ | $z_{\text{COK-2}} \leq y_{\text{COK}}$ |
| 'CC | $Y_{\text{FCC}} \Rightarrow Z_{\text{FCC-1}} \vee Z_{\text{FCC-2}}$ | $y_{\text{FCC}} \leq z_{\text{FCC-1}} + z_{\text{FCC-2}}$ |
| | $Z_{\text{FCC-1}} \Rightarrow Y_{\text{FCC}}$ | $z_{\text{FCC-1}} \leq y_{\text{FCC}}$ |
| | $Z_{\text{FCC-2}} \Rightarrow Y_{\text{FCC}}$ | $z_{\text{FCC-2}} \leq y_{\text{FCC}}$ |
| [CR | $Y_{\text{HCR}} \Rightarrow Z_{\text{HCR-1}} \vee Z_{\text{HCR-2}} \vee Z_{\text{HCR-3}} \vee Z_{\text{HCR-4}}$ | $y_{\text{HCR}} \leq z_{\text{HCR-1}} \vee z_{\text{HCR-2}} \vee z_{\text{HCR-3}} \vee z_{\text{HCR-4}}$ |
| | $Z_{\text{HCR-1}} \Rightarrow Y_{\text{HCR}}$ | $z_{\text{HCR-1}} \leq y_{\text{HCR}}$ |
| | $Z_{\text{HCR-2}} \Rightarrow Y_{\text{HCR}}$ | $z_{\text{HCR-2}} \leq y_{\text{HCR}}$ |
| | $Z_{\text{HCR-3}} \Rightarrow Y_{\text{HCR}}$ | $z_{\text{HCR-3}} \leq y_{\text{HCR}}$ |
| | $Z_{\text{HCR-4}} \Rightarrow Y_{\text{HCR}}$ | $z_{\text{HCR-4}} \leq y_{\text{HCR}}$ |

### 3.1.9 Mixed-integer linear programming (MILP) model formulation

The general model formulation for MILP is summarized as follows:

$$\min \quad Z = \sum_{u \in U} \mathrm{CAPEX}_u y_u + \sum_i \mathrm{OPEX}_i f_i \qquad (1)$$

s.t.

$$
\begin{array}{lll}
Ax = 0 & & \text{(material balances)} \\
F_i \le M y_i & \forall i \in I & \text{(big-}M\text{ logical constraints)} \\
Ay \le a & & \left(\begin{array}{l}\text{logical constraints on design specifications} \\ \text{and structural specifications}\end{array}\right) \\
x \in X \subseteq & & \text{(nonnegativity constraints on continuous variables)} \\
y \in \{0,1\}^q & & \text{(integrality constraints on binary variables)} \\
\mathrm{CAPEX}_u, \mathrm{OPEX}_i \ge 0 & &
\end{array}
$$

Also, according to Grossmann et al (1999), the superstructure is modeled mathematically as the following general formulation:

$$
\begin{array}{ll}
\min \mathrm{Z} = f(\boldsymbol{x}, \boldsymbol{y}) & \\
\text{s.t.} \quad \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}) = 0 & \\
\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}) \le 0 & \qquad (2) \\
\boldsymbol{x} \in \boldsymbol{X} \le \boldsymbol{R}^n & \\
\boldsymbol{y} \in \boldsymbol{Y} = \{0,1\}^l &
\end{array}
$$

where:

$f(x,y)$ is the objective function (cost minimization or profit maximization)

$h(x,y)=0$ are the $m$ equality constraints that describe the performance of the system (mass and heat balances, design equations)

$g(x,y) \le 0$ are the $p$ inequality constraints that define the specifications or constraints for feasible choices.

$x$ is a vector of $n$ continuous variables that correspond to the state or design variables flow rates, temperatures, pressures, composition etc. $\boldsymbol{R}^n$ is $n$ real numbers.

y is a vector of $l$ discrete variables which are restricted to 0-1 values to define the potential existence of a unit $i$ or an action

### 3.1.10 Genralized Disjunctive Programming (GDP) model formulation

In Turkay and Grossmann (1995), the GDP hybrid modeling formulation is given by:

$$\min Z = \sum_i c_i + f(x) + d^T y \qquad (3)$$

$$s.t.$$

$$\left.\begin{array}{l} g(x) \le 0 \\ r(x) + D(y) \le 0 \\ Ay \ge 0 \end{array}\right\} \qquad (4)$$

$$\begin{bmatrix} Y_i \\ h_i(x) \le 0 \\ c_i = \gamma_i \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ B^i x = 0 \\ c_i = 0 \end{bmatrix} \quad i \in D \qquad (5)$$

$$\Omega(Y) = True \qquad (6)$$

$$x \in R^n, c_i \ge 0, \mathrm{Y} \in \{\text{True, False}\}^m \qquad (7)$$

where:

$x$, $ci$: continuous variables

$y$: 0–1 variables,

$Yi$ : Boolean variables

$\Omega(Y)$: propositional logic involving only Boolean variables

$c_i \in \Re^m$: continuous variables of costs for each of the disjunctions that are activated to fixed charges of values $c_i$ if the corresponding term of the disjunction is true

$f$: $\Re^n \to \Re^1$: continuous variables $x$ in the objective function

$g$: $\Re^n \to \Re^q$: common constraint sets that hold regardless of the discrete decisions

$f(x)$ and $g(x)$ are convex functions

$g(x)$: linear or nonlinear inequality, independent of the discrete choices

$f(x)$: linear or nonlinear objective function

$r(x) + Dy \le 0$ : general mixed-integer algebraic equations

$Ay \ge a$: set of integer inequalities

$d^T y$ : linear cost terms

There are two approaches available (at least) for formulating the conditional constraints in a GDP model:

- Approach 1: direct formulation on the existence of the process units

$$
\text{ADU:}\begin{bmatrix} Y_{\text{ADU}} \\ (0.2088)\ f_{\text{CR}} = f_{\text{NAP1}} \\ (0.0555)\ f_{\text{CR}} = f_{\text{LSRN1}} \\ (0.1533)\ f_{\text{CR}} = f_{\text{HSRN1}} \\ f_{\text{NAP1}} = f_{\text{LSRN1}} + f_{\text{HSRN1}} \\ c_{\text{ADU}} = 228 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\text{ADU}} \\ f_{\text{CR}} = 0 \\ f_{\text{NAP1}} = 0 \\ f_{\text{LSRN1}} = 0 \\ f_{\text{HSRN1}} = 0 \\ c_{\text{ADU}} = 0 \end{bmatrix}
$$

- Approach 2: formulation by employing the logical constraints on design specifications:

$$
Y_{\text{ADU}} \Leftrightarrow \left(Y_{\text{HDT 1}} \vee Y_{\text{HDT 2}}\right)
$$

$$
\neg Y_{\text{ADU}} \vee \left(Y_{\text{HDT 1}} \vee Y_{\text{HDT 2}}\right)
$$

$$
\begin{bmatrix} \neg Y_{\text{ADU}} \\ f_{\text{NAP1}} = 0 \\ f_{\text{LSRN1}} = 0 \\ f_{\text{HSRN1}} = 0 \\ c_{\text{ADU}} = 0 \end{bmatrix} \vee \begin{bmatrix} Y_{\text{HDT 1}} \\ 0.0109\ (f_{\text{H2}} + f_{\text{HSRN 2}}) = f_{\text{FG 1}} \\ 0.0012\ (f_{\text{H2}} + f_{\text{HSRN 2}}) = f_{\text{H2S 1}} \\ 0.0058\ (f_{\text{H2}} + f_{\text{HSRN 2}}) = f_{\text{LPG 1}} \\ 0.2610\ (f_{\text{H2}} + f_{\text{HSRN 2}}) = f_{\text{LSRN 2}} \\ 0.7211\ (f_{\text{H2}} + f_{\text{HSRN 2}}) = f_{\text{HSRN 3}} \\ 0.9821\ (f_{\text{H2}} + f_{\text{HSRN 2}}) = f_{\text{NAP 4}} \\ c_{\text{HDT 1}} = 96 \end{bmatrix} \vee \begin{bmatrix} Y_{\text{HDT 2}} \\ 0.0109\ (f_{\text{H2}} + f_{\text{NAP 2}}) = f_{\text{FG 2}} \\ 0.0012\ (f_{\text{H2}} + f_{\text{NAP 2}}) = f_{\text{H2S 2}} \\ 0.0058\ (f_{\text{H2}} + f_{\text{NAP 2}}) = f_{\text{LPG 3}} \\ 0.2610\ (f_{\text{H2}} + f_{\text{NAP 2}}) = f_{\text{LSRN 3}} \\ 0.7211\ (f_{\text{H2}} + f_{\text{NAP 2}}) = f_{\text{HSRN 4}} \\ 0.9821\ (f_{\text{H2}} + f_{\text{NAP 2}}) = f_{\text{NAP 3}} \\ c_{\text{HDT 2}} = 96 \end{bmatrix}
$$

In this work, approach 1 is applied for the entire unit in the superstructure as listed in Table 3.6.

**Table 3.6:** GDP Direct formulation on process unit existence based on component material balances

| | |
|---|---|
| **ADU** | $\left[\begin{array}{c} Y_{\mathrm{ADU}} \\ (0.2088) f_{\mathrm{CR}} = f_{\mathrm{NAP\,1}} \\ (0.0555) f_{\mathrm{CR}} = f_{\mathrm{LSRN\,1}} \\ (0.1533) f_{\mathrm{CR}} = f_{\mathrm{HSRN\,1}} \\ f_{\mathrm{NAP\,1}} = f_{\mathrm{LSRN\,1}} + f_{\mathrm{HSRN\,1}} \\ c_{\mathrm{ADU}} = 228 \end{array}\right] \vee \left[\begin{array}{c} \neg Y_{\mathrm{ADU}} \\ f_{\mathrm{CR}} = 0 \\ f_{\mathrm{NAP\,1}} = 0 \\ f_{\mathrm{LSRN\,1}} = 0 \\ f_{\mathrm{HSRN\,1}} = 0 \\ c_{\mathrm{ADU}} = 0 \end{array}\right]$ |
| **HDT 1** | $\left[\begin{array}{c} Y_{\mathrm{HDT\,1}} \\ 0.0109\ (f_{\mathrm{H2}} + f_{\mathrm{HSRN\,2}}) = f_{\mathrm{FG\,1}} \\ 0.0012\ (f_{\mathrm{H2}} + f_{\mathrm{HSRN\,2}}) = f_{\mathrm{H2S\,1}} \\ 0.0058\ (f_{\mathrm{H2}} + f_{\mathrm{HSRN\,2}}) = f_{\mathrm{LPG\,1}} \\ 0.2610\ (f_{\mathrm{H2}} + f_{\mathrm{HSRN\,2}}) = f_{\mathrm{LSRN\,2}} \\ 0.7211\ (f_{\mathrm{H2}} + f_{\mathrm{HSRN\,2}}) = f_{\mathrm{HSRN\,3}} \\ 0.9821\ (f_{\mathrm{H2}} + f_{\mathrm{HSRN\,2}}) = f_{\mathrm{NAP\,4}} \\ c_{\mathrm{HDT\,1}} = 96 \end{array}\right] \vee \left[\begin{array}{c} \neg Y_{\mathrm{HDT\,1}} \\ f_{\mathrm{H2}} = f_{\mathrm{HSRN\,2}} = 0 \\ f_{\mathrm{FG\,1}} = 0 \\ f_{\mathrm{H2S\,1}} = 0 \\ f_{\mathrm{LPG1}} = 0 \\ f_{\mathrm{LSRN2}} = 0 \\ f_{\mathrm{HSRN3}} = 0 \\ f_{\mathrm{NAP4}} = 0 \\ c_{\mathrm{HDT\,1}} = 0 \end{array}\right]$ |
| **HDT 2** | $\left[\begin{array}{c} Y_{\mathrm{HDT\,2}} \\ 0.0109\ (f_{\mathrm{H2}} + f_{\mathrm{NAP\,2}}) = f_{\mathrm{FG\,2}} \\ 0.0012\ (f_{\mathrm{H2}} + f_{\mathrm{NAP\,2}}) = f_{\mathrm{H2S\,2}} \\ 0.0058\ (f_{\mathrm{H2}} + f_{\mathrm{NAP\,2}}) = f_{\mathrm{LPG\,3}} \\ 0.2610\ (f_{\mathrm{H2}} + f_{\mathrm{NAP\,2}}) = f_{\mathrm{LSRN\,3}} \\ 0.7211\ (f_{\mathrm{H2}} + f_{\mathrm{NAP\,2}}) = f_{\mathrm{HSRN\,4}} \\ 0.9821\ (f_{\mathrm{H2}} + f_{\mathrm{NAP\,2}}) = f_{\mathrm{NAP\,3}} \\ c_{\mathrm{HDT\,2}} = 96 \end{array}\right] \vee \left[\begin{array}{c} \neg Y_{\mathrm{HDT\,2}} \\ f_{\mathrm{H2}} = f_{\mathrm{NAP\,2}} = 0 \\ f_{\mathrm{FG2}} = 0 \\ f_{\mathrm{H2S2}} = 0 \\ f_{\mathrm{LPG3}} = 0 \\ f_{\mathrm{LSRN3}} = 0 \\ f_{\mathrm{HSRN4}} = 0 \\ f_{\mathrm{NAP3}} = 0 \\ c_{\mathrm{HDT\,2}} = 0 \end{array}\right]$ |
| **ISO** | $\left[\begin{array}{c} Y_{\mathrm{ISO}} \\ (0.9900)\ f_{\mathrm{LSRN\,5}} = f_{\mathrm{ISO}} \\ (0.0100)\ f_{\mathrm{LSRN\,5}} = f_{\mathrm{FG\,4}} \\ c_{\mathrm{ISO}} = 42 \end{array}\right] \vee \left[\begin{array}{c} \neg Y_{\mathrm{ISO}} \\ f_{\mathrm{LSRN\,5}} = 0 \\ f_{\mathrm{ISO}} = 0 \\ f_{\mathrm{FG4}} = 0 \\ c_{\mathrm{ISO}} = 0 \end{array}\right]$ |
| **SRU** | $\left[\begin{array}{c} Y_{\mathrm{SRU}} \\ 0.8478\ (f_{\mathrm{H2S\,1}} + f_{\mathrm{H2S\,2}}) = f_{\mathrm{S}} \\ 0.1522\ (f_{\mathrm{H2S\,1}} + f_{\mathrm{H2S\,2}}) = f_{\mathrm{TG}} \\ c_{\mathrm{SRU}} = 30 \end{array}\right] \vee \left[\begin{array}{c} \neg Y_{\mathrm{SRU}} \\ f_{\mathrm{H2S\,1}} = f_{\mathrm{H2S\,2}} = 0 \\ f_{\mathrm{S}} = 0 \\ f_{\mathrm{TG}} = 0 \\ c_{\mathrm{SRU}} = 0 \end{array}\right]$ |
| **REF** | $\left[\begin{array}{c} Y_{\mathrm{REF}} \\ 0.0320\ (f_{\mathrm{HSRN\,5}} + f_{\mathrm{NAP\,5}}) = f_{\mathrm{H2}} \\ 0.0370\ (f_{\mathrm{HSRN\,5}} + f_{\mathrm{NAP\,5}}) = f_{\mathrm{FG\,3}} \\ 0.0780\ (f_{\mathrm{HSRN\,5}} + f_{\mathrm{NAP\,5}}) = f_{\mathrm{LPG\,2}} \\ 0.8530\ (f_{\mathrm{HSRN\,5}} + f_{\mathrm{NAP\,5}}) = f_{\mathrm{REF}} \\ c_{\mathrm{REF}} = 270 \end{array}\right] \vee \left[\begin{array}{c} \neg Y_{\mathrm{REF}} \\ f_{\mathrm{HSRN\,5}} = f_{\mathrm{NAP\,5}} = 0 \\ f_{\mathrm{H2}} = 0 \\ f_{\mathrm{FG3}} = 0 \\ f_{\mathrm{LPG2}} = 0 \\ f_{\mathrm{REF}} = 0 \\ c_{\mathrm{REF}} = 0 \end{array}\right]$ |

| | |
|---|---|
| **SOLD** | $\begin{bmatrix} Y_{\mathrm{SOLD}} \\ f_{\mathrm{SOLD}} = f_{\mathrm{LSRN\,6}} + f_{\mathrm{S}} + f_{\mathrm{GSLN}} + f_{\mathrm{LPG\,5}} \\ c_{\mathrm{SOLD}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{SOLD}} \\ f_{\mathrm{SOLD}} = f_{\mathrm{LSRN\,6}} = f_{\mathrm{S}} = f_{\mathrm{GSLN}} = f_{\mathrm{LPG\,5}} = 0 \\ c_{\mathrm{SOLD}} = 0 \end{bmatrix}$ |
| **BLND** | $\begin{bmatrix} Y_{\mathrm{GSLN}} \\ f_{\mathrm{GSLN}} = f_{\mathrm{ISO}} + f_{\mathrm{REF}} \\ c_{\mathrm{GSLN}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{GSLN}} \\ f_{\mathrm{GSLN}} = f_{\mathrm{ISO}} = f_{\mathrm{REF}} = 0 \\ c_{\mathrm{GSLN}} = 0 \end{bmatrix}$ |
| **LPG** | $\begin{bmatrix} Y_{\mathrm{LPG}} \\ f_{\mathrm{LPG5}} = f_{\mathrm{LPG4}} \\ c_{\mathrm{LPG}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{LPG}} \\ f_{\mathrm{LPG5}} = f_{\mathrm{LPG4}} = 0 \\ c_{\mathrm{LPG}} = 0 \end{bmatrix}$ |
| **FGH** | $\begin{bmatrix} Y_{\mathrm{FG}} \\ f_{FG5} = f_{\mathrm{FG1}} + f_{\mathrm{FG2}} + f_{\mathrm{FG3}} + f_{\mathrm{FG4}} \\ c_{\mathrm{FG}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{FG}} \\ f_{\mathrm{FG5}} = f_{\mathrm{FG1}} = f_{\mathrm{FG2}} = f_{\mathrm{FG3}} = f_{\mathrm{FG4}} = 0 \\ c_{\mathrm{FG}} = 0 \end{bmatrix}$ |
| **SPLT 1** | $\begin{bmatrix} Y_{\mathrm{SPLT\,1}} \\ (0.9000)\ \ f_{\mathrm{LSRN\,4}} = f_{\mathrm{LSRN\,5}} \\ (0.1000)\ \ f_{\mathrm{LSRN\,4}} = f_{\mathrm{LSRN\,6}} \\ c_{\mathrm{SPLT\,1}} = \alpha \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{SPLT\,1}} \\ f_{\mathrm{LSRN\,4}} = f_{\mathrm{LSRN5}} = f_{\mathrm{LSRN6}} = 0 \\ c_{\mathrm{SPLT\,1}} = 0 \end{bmatrix}$ |
| **SPLT 2** | $\begin{bmatrix} Y_{\mathrm{SPLT\,2}} \\ f_{\mathrm{H2}} = f_{\mathrm{H2\_1}} + f_{\mathrm{H2\_2}} \\ c_{\mathrm{SPLT\,2}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{SPLT\,2}} \\ f_{\mathrm{H2}} = f_{\mathrm{H2\_1}} = f_{\mathrm{H2\_2}} = 0 \\ c_{\mathrm{SPLT\,2}} = 0 \end{bmatrix}$ |
| **MIX 1** | $\begin{bmatrix} Y_{\mathrm{MIX\,1}} \\ f_{\mathrm{NAP1}} + f_{\mathrm{VIS\_2}} + f_{\mathrm{COK\_2}} + f_{\mathrm{FCC\_2}} + f_{\mathrm{HCR\_2}} + f_{\mathrm{PCHN1\_2}} = f_{\mathrm{NAP2}} \\ c_{\mathrm{MIX\,1}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{MIX\,1}} \\ f_{\mathrm{NAP1}} = f_{\mathrm{VIS\_2}} = f_{\mathrm{COK\_2}} = f_{\mathrm{FCC\_2}} = 0 \\ f_{\mathrm{HCR\_2}} = f_{\mathrm{PCHN1\_2}} = f_{\mathrm{NAP2}} = 0 \\ c_{\mathrm{MIX\,1}} = 0 \end{bmatrix}$ |
| **MIX 2** | $\begin{bmatrix} Y_{\mathrm{MIX\,2}} \\ f_{\mathrm{HSRN1}} + f_{\mathrm{VIS\_1}} + f_{\mathrm{COK\_1}} + f_{\mathrm{FCC\_1}} + f_{\mathrm{HCR\_1}} + f_{\mathrm{PCHN1\_1}} = f_{\mathrm{HSRN2}} \\ c_{\mathrm{MIX\,2}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{MIX\,2}} \\ f_{\mathrm{HSRN1}} = f_{\mathrm{VIS\_1}} = f_{\mathrm{COK\_1}} = f_{\mathrm{FCC\_1}} = 0 \\ f_{\mathrm{HCR\_1}} = f_{\mathrm{PCHN1\_1}} = f_{\mathrm{HSRN2}} = 0 \\ c_{\mathrm{MIX\,2}} = 0 \end{bmatrix}$ |
| **MIX 3** | $\begin{bmatrix} Y_{\mathrm{MIX\,3}} \\ f_{\mathrm{LSRN1}} + f_{\mathrm{LSRN2}} + f_{\mathrm{LSRN3}} + f_{\mathrm{PCHN2}} = f_{\mathrm{LSRN4}} \\ c_{\mathrm{MIX\,3}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{MIX\,3}} \\ f_{\mathrm{LSRN1}} = f_{\mathrm{LSRN2}} = f_{\mathrm{LSRN3}} = f_{\mathrm{PCHN2}} = f_{\mathrm{LSRN4}} = 0 \\ c_{\mathrm{MIX\,3}} = 0 \end{bmatrix}$ |
| **MIX 4** | $\begin{bmatrix} Y_{\mathrm{MIX\,4}} \\ f_{\mathrm{HSRN3}} + f_{\mathrm{HSRN4}} + f_{\mathrm{PCHN3\_1}} + f_{\mathrm{HCR\_3}} = f_{\mathrm{HSRN5}} \\ c_{\mathrm{MIX\,4}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{MIX\,4}} \\ f_{\mathrm{HSRN3}} = f_{\mathrm{HSRN4}} = f_{\mathrm{PCHN3\_1}} = f_{\mathrm{HCR\_3}} = f_{\mathrm{HSRN5}} = 0 \\ c_{\mathrm{MIX\,4}} = 0 \end{bmatrix}$ |
| **MIX 5** | $\begin{bmatrix} Y_{\mathrm{MIX\,5}} \\ f_{\mathrm{NAP3}} + f_{\mathrm{NAP4}} + f_{\mathrm{PCHN3\_2}} + f_{\mathrm{HCR\_4}} = f_{\mathrm{NAP5}} \\ c_{\mathrm{MIX\,5}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{MIX\,5}} \\ f_{\mathrm{NAP3}} = f_{\mathrm{NAP4}} = f_{\mathrm{PCHN3\_2}} = f_{\mathrm{HCR\_4}} = f_{\mathrm{NAP5}} = 0 \\ c_{\mathrm{MIX\,5}} = 0 \end{bmatrix}$ |
| **MIX 6** | $\begin{bmatrix} Y_{\mathrm{MIX\,6}} \\ f_{\mathrm{LPG1}} + f_{\mathrm{LPG2}} + f_{\mathrm{LPG3}} = f_{\mathrm{LPG4}} \\ c_{\mathrm{MIX\,6}} = 10 \end{bmatrix} \vee \begin{bmatrix} \neg Y_{\mathrm{MIX\,6}} \\ f_{\mathrm{LPG1}} = f_{\mathrm{LPG2}} = f_{\mathrm{LPG3}} = f_{\mathrm{LPG4}} = 0 \\ c_{\mathrm{MIX\,6}} = 0 \end{bmatrix}$ |

### 3.1.11 Systematic transformation from logical proposition into mathematical representation

According to Raman and Grossmann (1994), the three steps procedures to transform each logical proposition into equivalent mathematical representation in terms of linear equality/inequality by using binary (0-1) variables are:

1) replace the implication by its equivalent disjunction:

   $P_1 \Rightarrow P_2 \Leftrightarrow \neg P_1 \vee P_2$ (8)

2) move the negation inward by applying DeMorgan's Theorem:

   $\neg(P_1 \wedge P_2) \Leftrightarrow \neg P_1 \vee \neg P_2$ (9) $\neg(P_1 \vee P_2) \Leftrightarrow \neg P_1 \wedge \neg P_2$ (10)

3) recursively distribute the "OR' over the "AND":

   $(P_1 \wedge P_2) \vee P_3 \Leftrightarrow (P_1 \vee P_2) \wedge (P_1 \vee P_3)$ (11)

where:

$P_i$, : proposition, given by binary variable $y_i$

$\neg P_i$ : proposition negation or complement, given by $1 - y_i$

A proposition is any logical expression and consists of a set of clauses *Pi*, *i*=1 are related by the logical operators OR, AND, IMPLICATION. Having converted each logical proposition into its conjunctive normal form representation, $Q_1 \wedge Q_2 \wedge ... \wedge Q_S$, it can then be easily expressed as a set of linear equality and inequality constraints.

## 3.2 PROJECT ACTIVITIES

The proposed methodology to address the conceptual design or process synthesis problem of the naphtha produced from the ADU is presented in this chapter.

In general, the mathematical programming approach to process synthesis and design activities and problems consists of the following four major steps (Yeomans and Grossmann, 1999) with the following descriptions:

1. Development of the superstructure to represent the space of topological alternatives of the naphtha flow to petrochemical plant configuration;
2. Establishment of the general solution strategy to determine the optimal topology from the superstructure representation of candidates. Simultaneous optimization strategy is employed instead of sequential optimization strategy due to the model is largely linear;
3. Formulation or modeling of the postulated superstructure in a mathematical form that involves discrete and continuous variables for the selection of the configuration and operating levels, respectively. In this model, mixed integer linear programming (MILP) and generalized disjunctive programming (GDP) are being modeled; and
4. Solution of the corresponding mathematical form, i.e., the optimization model from which the optimal topology is determined, in which to solve the MILP optimization model using Cplex solver and GDP formulation using LOGMIP solver within the GAMS modeling language environment.

A diagrammatic description of methodology is shown in Figure 3.2. Project milestone and Gantt chart are shown in Table 3.7 and Table 3.8.

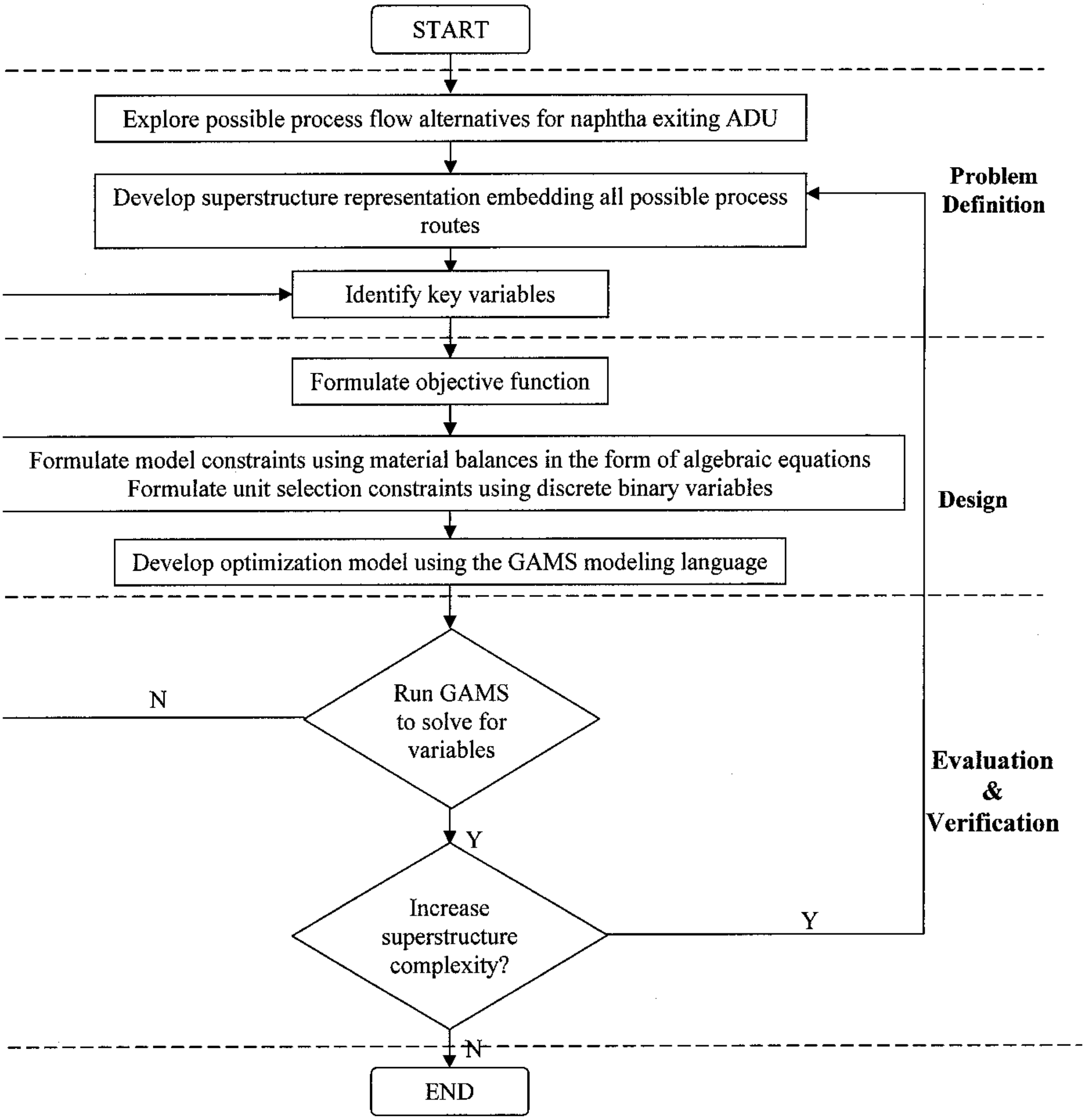


**Figure 3.2:** Flow chart of the proposed methodology to carry out the thesis research

## 3.3 GANTT CHART

**Table 3.7:** Gantt Chart for FYP I

| **Details/Week** | **FYP I** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | | **10** | **11** | **12** | **13** | **14** |
| Problem identification | √ | √ | | | | | | | | | | | | | |
| Literature review | | √ | √ | | | | | | | | | | | | |
| Development of design and structural logical constraints | | | | √ | √ | √ | | | | | | | | | |
| MILP formulation | | | | | | √ | √ | √ | √ | | | | | | |
| GDP formulation | | | | | | | | | √ | | √ | | | | |
| Solve MILP optimization model using GAMS | | | | | | | | | | | | √ | √ | | |
| Submission of interim report and oral presentation | | | | | | | | | | | | | | √ | √ |

**Table 3.8:** Gantt Chart for FYP II

| **Details/Week** | **FYP II** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | | **10** | **11** | **12** | **13** | **14** |
| Solve MILP model in GAMS/Cplex | √ | √ | √ | √ | | | | | | | | | | | |
| Solve GDP model in GAMS/LOGMIP | | | | √ | √ | √ | | | | | | | | | |
| Result comparison between MILP and GDP | | | | | | | √ | | | | | | | | |
| Result verification | | | | | | | | √ | √ | | | | | | |
| PreEDX poster presentation | | | | | | | | | | | √ | | | | |
| Interim report submission | | | | | | | | | | | | √ | √ | √ | |
| Final oral presentation and hardbound submission | | | | | | | | | | | | | | | √ |

## 3.4 COMPUTATIONAL TOOLS

Computational experiments and numerical studies of the MILP model formulation for the flowsheet superstructure optimization problem proposed in this work are coded and implemented using GAMS 22.8 under Integrated Development Environment (IDE) version for Windows Vista Home Basic platform. The numerical examples are solved using the solver GAMS/CPLEX 10 (ILOG Cplex Division, last retrieved April 28, 2009) on an Intel(R) Centrino Core(TM) 2 Duo, 2.0GHz, 4.0GB of RAM Toshiba notebook.

# CHAPTER 4

# COMPUTATIONAL EXPERIENCE AND NUMERICAL RESULTS

## 4.1 COMPUTATIONAL EXPERIMENTS

The model computational results and discussion can be divided into two parts, namely: mixed-integer linear programming (MILP) and generalized disjunctive programming (GDP). The associated model and computational statistic are reported in Table 4.1 for MILP and Table 4.2 for GDP. The MILP and GAMS code are attached in Appendix A and Appendix B respectively, while the complete computational result for MILP and GDP are in Appendix C and Appendix D respectively. MILP model is solved by using GAMS/Cplex 10 solver, which gives a slightly longer CPU time/resource usage of 0.053s, as compared to the CPU time/resource usage of 0.037s for GDP model which solved by using GAMS/LOGMIP solver. However, the CPU time/resource usage for both model is trivial, and the differences are insignificant. This is due to the model is relatively small, in which the effect on CPU time/resource usage is negligible. By comparing statistic from both of the models, it is concluded that MILP model requires more number of single equation, binary variables and iterations as compared to GDP model. The result agrees with the advantages of GDP stated by Yeomans and Grossmann (1999), in which binary 0 – 1 variables are explicitly included in the model. The model size will be reduced by only considering disjunctions for which Boolean variable is true, instead of heavy combinatorial search effort of binary variables.

**Table 4.1:** Model and computational statistics (MILP)

| Solver | **CPLEX 10** |
|---|---|
| Number of single equations | **336** |
| Number of binary variables | **80** |
| Number of iterations | **26** |
| CPU time/resource usage | **0.053 s** |

**Table 4.2:** Model and computational statistics (GDP)

| Solver | **CPLEX 10** |
|---|---|
| Number of single equations | **331** |
| Number of binary variables | **75** |
| Number of iterations | **22** |
| CPU time/resource usage | **0.037s** |

For each type of model, two design scenarios as distinguished by the API gravity (specific gravity) of the crude charge to the ADU are considered in the computational experiments conducted on the proposed modeling approach, namely light crude charge processing (API > 33) and heavy crude charge processing (API ≤ 33). The corresponding model to be solved is stipulated through the use of if–else selection statement in the GAMS implementation. Table 4.3 shows the utilities cost per unit and Table 4.4 shows the base cost and utilities consumption of major unit operations (Maples, 2000, p.386)

## 4.2 **BASE DATA**

1. Input data from user (the input values are in brackets):
   a. Production requirement of gasoline (100,000kg/d)
   b. Crude API gravity (44.6)
   c. Crude oil cost (RM 120.0 per bbl)
   d. Naphtha cost (RM 0.524 per kg)

2. Nelson–Farrar Refinery Construction Index (NFRCI) (Maples, 2000, p. 388; EU-OPEC Roundtable on Energy Policies, 2008):
   a. Jan 1991 : 1241.7
   b. Dec 2008 : 2067.2

3. Assumptions:
   a. Refinery operates 330 days per year
   b. Crude charge is fixed to be between 100 000 bbl/d to 150 000 bbl/d
   c. Gasoline requirement of at least 100 000 kg/d
   d. Maximum capacity of each unit = $1 \times 10^8$ kg
   e. Total capital investment
      = Fixed capital investment + Working capital
      = Total equipment base cost + Working capital
   f. Total operating cost
      = Fixed operating costs + Variable operating costs + General expenses
   g. Total cost (objective function)
      = Total capital investment + Total operating cost

4. Utilities cost:

**Table 4.3:** Utilities cost per unit (www.mida.gov.my)(2008)

| Utilities Cost per Unit (RM/unit) | |
|---|---|
| Electricity (per kWh) | 0.1980 |
| Fuel (per MJ) | 0.1018 |
| HP Steam (per kg) | 0.0050 |
| CW (per $m^3$) | 0.8400 |

5. Base Capacity, Base Cost and Utilities Consumption of Major Unit Operations:

**Table 4.4:** Base cost and utilities consumption of major unit operations (Maples, 2000, p.386)

| | | | Utilities Consumption (Light crude) | | | |
|---|---|---|---|---|---|---|
| **Process** | **Jan '91 (mil RM)** | **Dec '08 (mil RM)** | **Electricity (MWh/kg)** | **Fuel (kJ/kg)** | **Steam (kg/kg)** | **CW (m3/kg)** |
| ADU | 137 | 228 | 0.0039 | 0.0826 | 0.0888 | 0.0000 |
| VIS | 86 | 144 | 0.0039 | 0.0660 | 0.1776 | 0.0000 |
| COK | 166 | 276 | 0.0282 | 0.0991 | 0.1421 | 0.0000 |
| FCC | 310 | 515 | 0.0078 | 0.0660 | 0.0710 | 0.0119 |
| HCR | 342 | 569 | 0.1402 | 0.2766 | 0.0000 | 0.0000 |
| HDT | 58 | 96 | 0.0157 | 0.0248 | 0.0533 | 0.0000 |
| REF | 162 | 270 | 0.0078 | 0.2477 | 0.1421 | 0.0030 |
| ISO | 25 | 42 | 0.0078 | 0.0083 | 0.1279 | 0.0000 |
| SRU (per tonne) | 18 | 30 | 0.3132 | 0.0000 | 2.6636 | 0.1482 |

## 4.3 COMPUTATIONAL RESULTS FOR MILP

### 4.3.1 Optimizer for crude feed requirement

**Input data:**

- total feed flowrate from units in the refinery varies dependent on light or heavy crude (feed from VIS, COK, FCC, HCR are constant at 2000000 kg/d)
- production requirements are constant.

**Table 4.5:** Computational results on the cost components for MILP crude feed dependent

| | Light Crude | Heavy Crude |
|---|---|---|
| CAPEX + OPEX + Raw material (*mil RM*) | 2744 | 2743 |
| CAPEX (*mil RM*) | 791 | 791 |
| OPEX + Raw material (*mil RM*) | 1953 | 1951 |
| Crude feed requirement (*kg/d*) | 4.5E+7 | 3.2E+7 |
| Raw material (*mil RM*) | 42 | 30 |
| OPEX (*mil RM*) | 1910 | 1921 |
| OPEX (*mil RM/d*) | 5.7 | 5.8 |

Note: assuming 330 working days

**Table 4.6:** Computational results on the stream flow rates for MILP crude feed dependent (light crude)

| Stream | Flowrate (kg/d) | Stream | Flow rate (kg/d) | Stream | Flow rate (kg/d) |
|---|---|---|---|---|---|
| COK_1 | 0 | HCR_4 | 0 | NAP1 | 0 |
| COK_2 | 0 | HSRN1 | 6958923 | NAP2 | 2.2E+7 |
| FCC_1 | 2000000 | HSRN2 | 1.0E+7 | NAP3 | 0 |
| FCC_2 | 0 | HSRN3 | 4235493 | NAP4 | 0 |
| FG1 | 127463 | HSRN4 | 0 | NAP5 | 2.2E+7 |
| FG2 | 0 | HSRN5 | 0 | PCHN1_1 | 0 |
| FG3 | 849861 | ISO | 2244763 | PCHN1_2 | 0 |
| FG4 | 22674 | LPG1 | 67824 | PCHN2 | 0 |
| FG5 | 1000000 | LPG2 | 1791600 | PCHN3_1 | 0 |
| GSLN | 2.1E+7 | LPG3 | 0 | PCHN3_2 | 0 |
| H2 | 735015 | LPG4 | 1859425 | REF | 1.9E+7 |
| H2_1 | 735015 | LPG5 | 1859425 | S | 11896 |
| H2_2 | 0 | LSRN1 | 2519375 | SOLD | 2.3E+7 |
| H2S1 | 14032 | LSRN2 | 0 | TG | 2135 |
| H2S2 | 0 | LSRN3 | 0 | VIS_1 | 2000000 |
| HCR_1 | 0 | LSRN4 | 2519375 | VIS_2 | 0 |
| HCR_2 | 0 | LSRN5 | 2267437 | CR | 4.5E+7 |
| HCR_3 | 0 | LSRN6 | 251937 | | |

**Table 4.7:** Computational results on the stream flow rates for MILP crude feed dependent (heavy crude)

| Stream | Flowrate (kg/d) | Stream | Flow rate (kg/d) | Stream | Flow rate (kg/d) |
|---|---|---|---|---|---|
| COK_1 | 2000000 | HCR_4 | 0 | NAP1 | 0 |
| COK_2 | 0 | HSRN1 | 5029419 | NAP2 | 0 |
| FCC_1 | 2000000 | HSRN2 | 1.1E+7 | NAP3 | 0 |
| FCC_2 | 0 | HSRN3 | 0 | NAP4 | 2.3E+7 |
| FG1 | 128283 | HSRN4 | 0 | NAP5 | 2.3E+7 |
| FG2 | 0 | HSRN5 | 0 | PCHN1_1 | 0 |
| FG3 | 855328 | ISO | 1622356 | PCHN1_2 | 0 |
| FG4 | 16387 | LPG1 | 68261 | PCHN2 | 0 |
| FG5 | 1000000 | LPG2 | 1803125 | PCHN3_1 | 0 |
| GSLN | 2.1E+7 | LPG3 | | PCHN3_2 | 0 |
| H2 | 739743 | LPG4 | 1871386 | REF | 1.9E+7 |
| H2_1 | 739743 | LPG5 | 1871386 | S | 11973 |
| H2_2 | 0 | LSRN1 | 1820827 | SOLD | 2.3E+7 |
| H2S1 | 14122 | LSRN2 | 0 | TG | 2149 |
| H2S2 | 0 | LSRN3 | 0 | VIS_1 | 0 |
| HCR_1 | 2000000 | LSRN4 | 1820827 | VIS_2 | 0 |
| HCR_2 | 0 | LSRN5 | 1638744 | CR | 3.2E+7 |
| HCR_3 | 0 | LSRN6 | 182082 | | 0 |

### 4.3.2 Optimizer for operating cost

**Input data:**

- total feed flowrate from units in the refinery constant, non-dependent on light or heavy crude (feed from VIS, COK, HCR are set at 2000000 kg/d while from FCC set at 3000000 kg/d)
- production requirements are constant.

**Table 4.8:** Computational results on the cost components for MILP operating cost dependent

| | Light Crude | Heavy Crude |
|---|---|---|
| CAPEX + OPEX + Raw material (*mil RM*) | 2688 | 2743 |
| CAPEX (*mil RM*) | 791 | 791 |
| OPEX + Raw material (*mil RM*) | 1897 | 1951 |
| Crude feed requirement (*kg/d*) | 3.28E+7 | 3.28E+7 |
| Raw material (*mil RM*) | 30 | 30 |
| OPEX (*mil RM*) | 1866 | 1921 |
| OPEX (*mil RM/d*) | 5.6 | 5.8 |

Note: assuming 330 working days

**Table 4.9:** Computational results on the stream flow rates for MILP operating cost dependent (light crude)

| Stream | Flowrate (kg/d) | Stream | Flow rate (kg/d) | Stream | Flow rate (kg/d) |
|---|---|---|---|---|---|
| COK_1 | 0 | HCR_4 | 0 | NAP1 | 0 |
| COK_2 | 0 | HSRN1 | 5029419 | NAP2 | 0 |
| FCC_1 | 3000000 | HSRN2 | 1.10E+7 | NAP3 | 0 |
| FCC_2 | 0 | HSRN3 | 0 | NAP4 | 2.31E+7 |
| FG1 | 128283 | HSRN4 | 0 | NAP5 | 2.31E+7 |
| FG2 | 0 | HSRN5 | 0 | PCHN1_1 | 0 |
| FG3 | 855328 | ISO | 1622356 | PCHN1_2 | 0 |
| FG4 | 16387 | LPG1 | 68261 | PCHN2 | 0 |
| FG5 | 1000000 | LPG2 | 1803125 | PCHN3_1 | 0 |
| GSLN | 2.13E+7 | LPG3 | | PCHN3_2 | 0 |
| H2 | 739743 | LPG4 | 1871386 | REF | 1.97E+7 |
| H2_1 | 739743 | LPG5 | 1871386 | S | 11973 |
| H2_2 | 0 | LSRN1 | 1820827 | SOLD | 2.34E+7 |
| H2S1 | 14122 | LSRN2 | 0 | TG | 2149 |
| H2S2 | 0 | LSRN3 | 0 | VIS_1 | 3000000 |
| HCR_1 | 0 | LSRN4 | 1820827 | VIS_2 | 0 |
| HCR_2 | 0 | LSRN5 | 1638744 | CR | 3.28E+7 |
| HCR_3 | 0 | LSRN6 | 182082 | | 0 |

**Table 4.10:** Computational results on the stream flow rates for MILP operating cost dependent (heavy crude)

| Stream | Flowrate (kg/d) | Stream | Flow rate (kg/d) | Stream | Flow rate (kg/d) |
|---|---|---|---|---|---|
| COK_1 | 2000000 | HCR_4 | 0 | NAP1 | 0 |
| COK_2 | 0 | HSRN1 | 5029419 | NAP2 | 0 |
| FCC_1 | 2000000 | HSRN2 | 1.1E+7 | NAP3 | 0 |
| FCC_2 | 0 | HSRN3 | 0 | NAP4 | 2.3E+7 |
| FG1 | 128283 | HSRN4 | 0 | NAP5 | 2.3E+7 |
| FG2 | 0 | HSRN5 | 0 | PCHN1_1 | 0 |
| FG3 | 855328 | ISO | 1622356 | PCHN1_2 | 0 |
| FG4 | 16387 | LPG1 | 68261 | PCHN2 | 0 |
| FG5 | 1000000 | LPG2 | 1803125 | PCHN3_1 | 0 |
| GSLN | 2.1E+7 | LPG3 | | PCHN3_2 | 0 |
| H2 | 739743 | LPG4 | 1871386 | REF | 1.9E+7 |
| H2_1 | 739743 | LPG5 | 1871386 | S | 11973.476 |
| H2_2 | 0 | LSRN1 | 1820827 | SOLD | 2.3E+7 |
| H2S1 | 14122 | LSRN2 | 0 | TG | 2149 |
| H2S2 | 0 | LSRN3 | 0 | VIS_1 | 0 |
| HCR_1 | 2000000 | LSRN4 | 1820827 | VIS_2 | 0 |
| HCR_2 | 0 | LSRN5 | 1638744 | CR | 3.2E+7 |
| HCR_3 | 0 | LSRN6 | 182082 | | 0 |

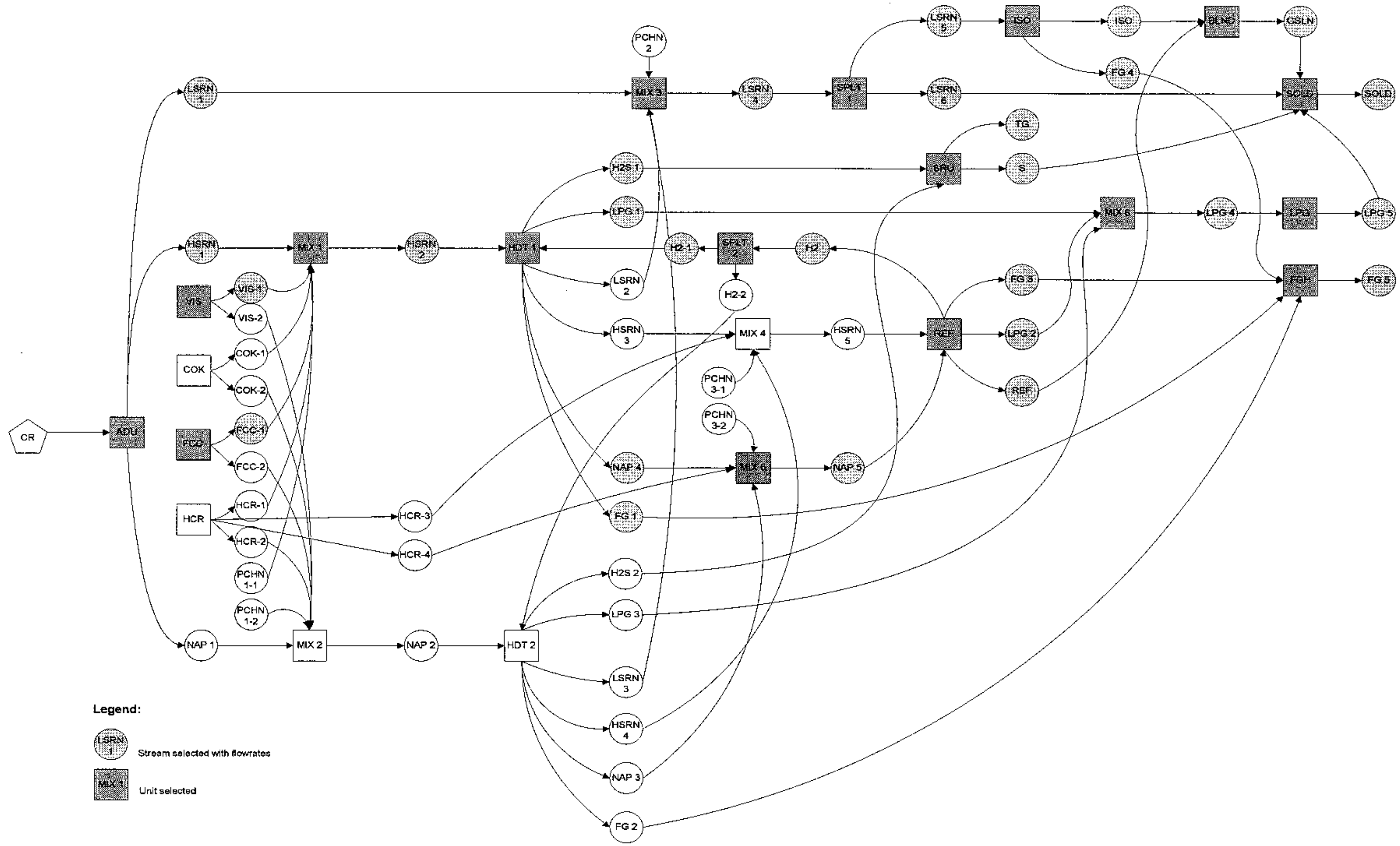


**Figure 4.1:** Optimal superstructure topology naphtha produced from ADU with light crude charge For crude feed dependent and operating cost dependent (MILP)

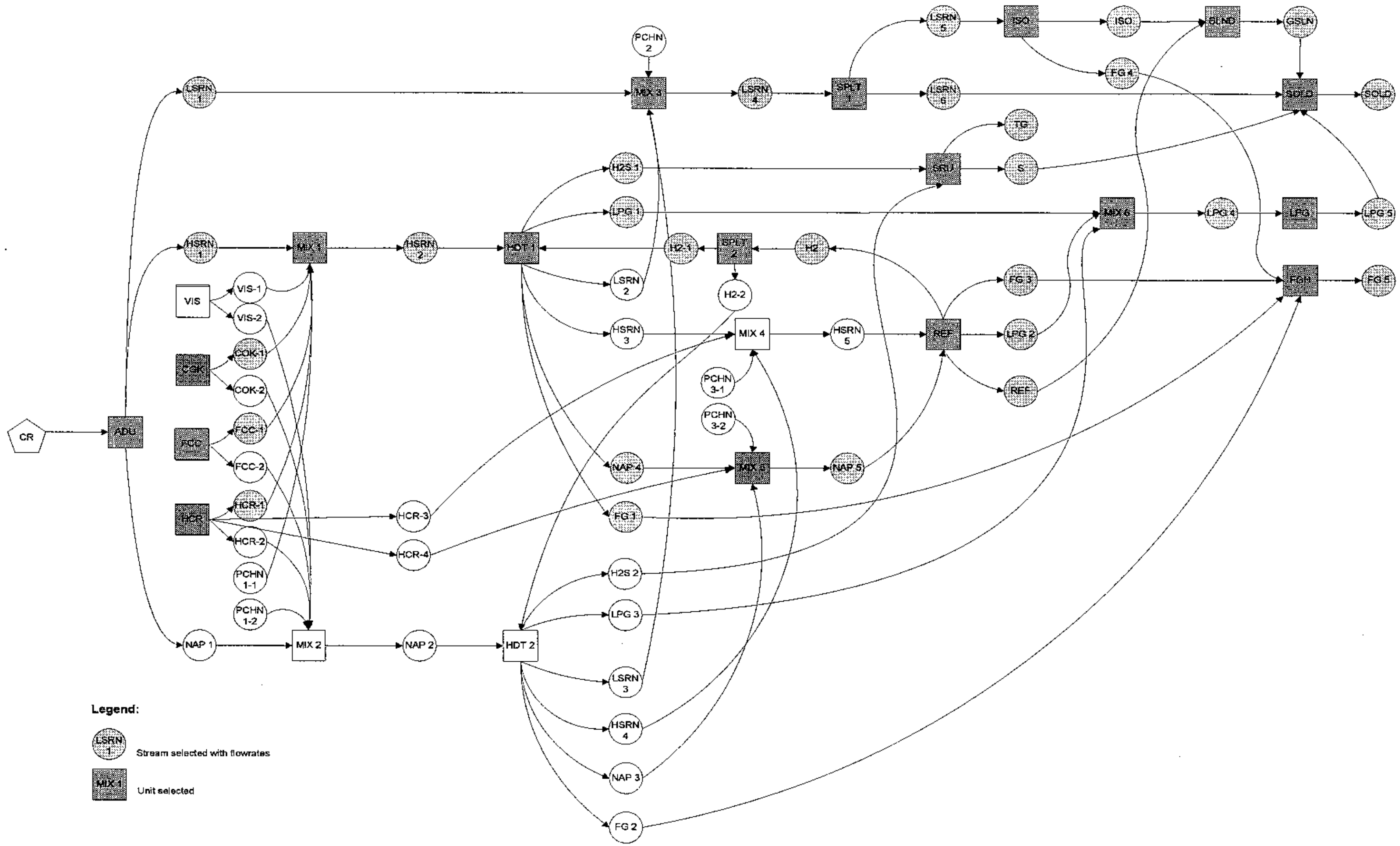


**Figure 4.2:** Optimal superstructure topology naphtha produced from ADU with heavy crude charge

For crude feed dependent and operating cost dependent (MILP)

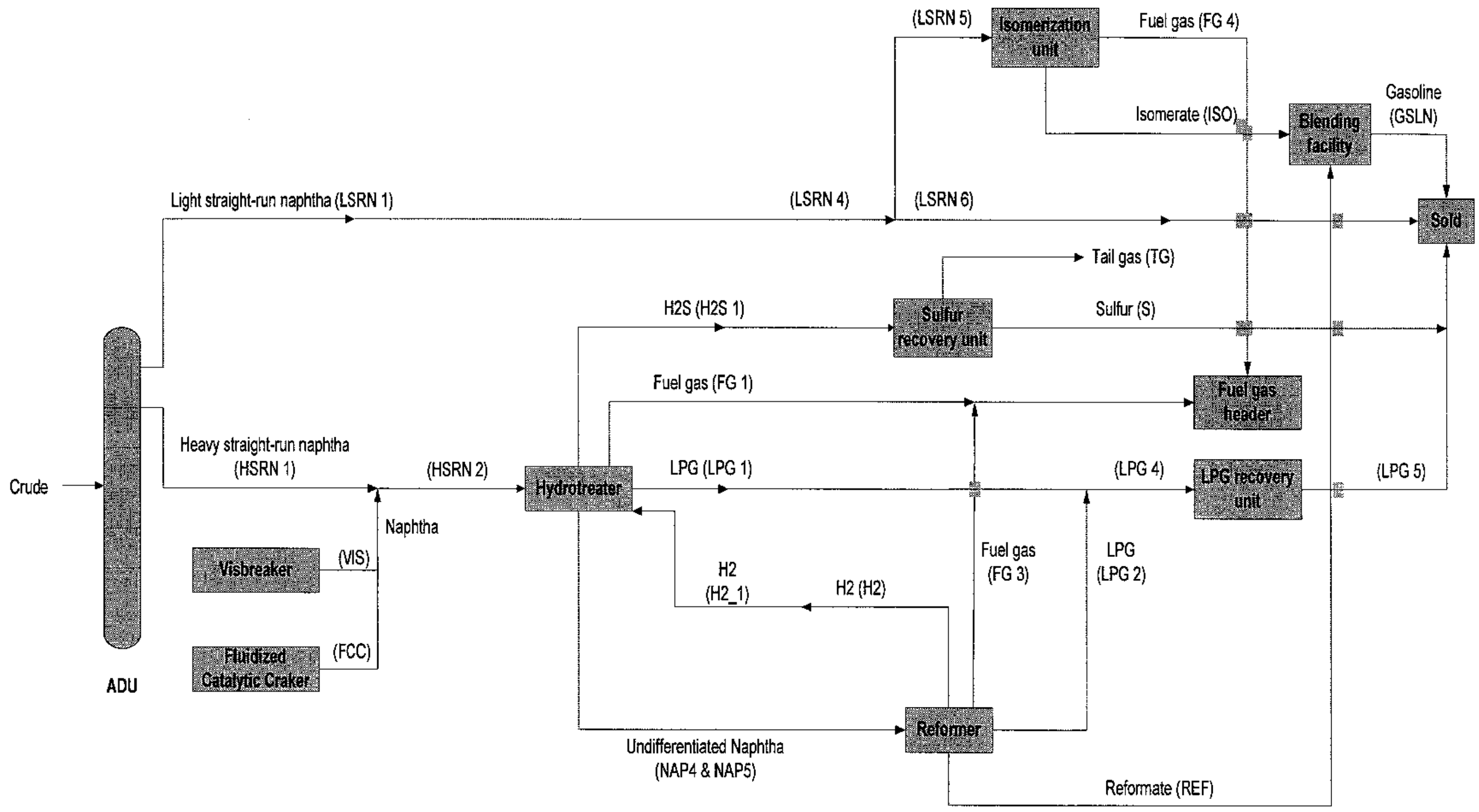


**Figure 4.3:** Optimal flowsheet for naphtha with light crude charge API > 33° and the specified input data (MILP)

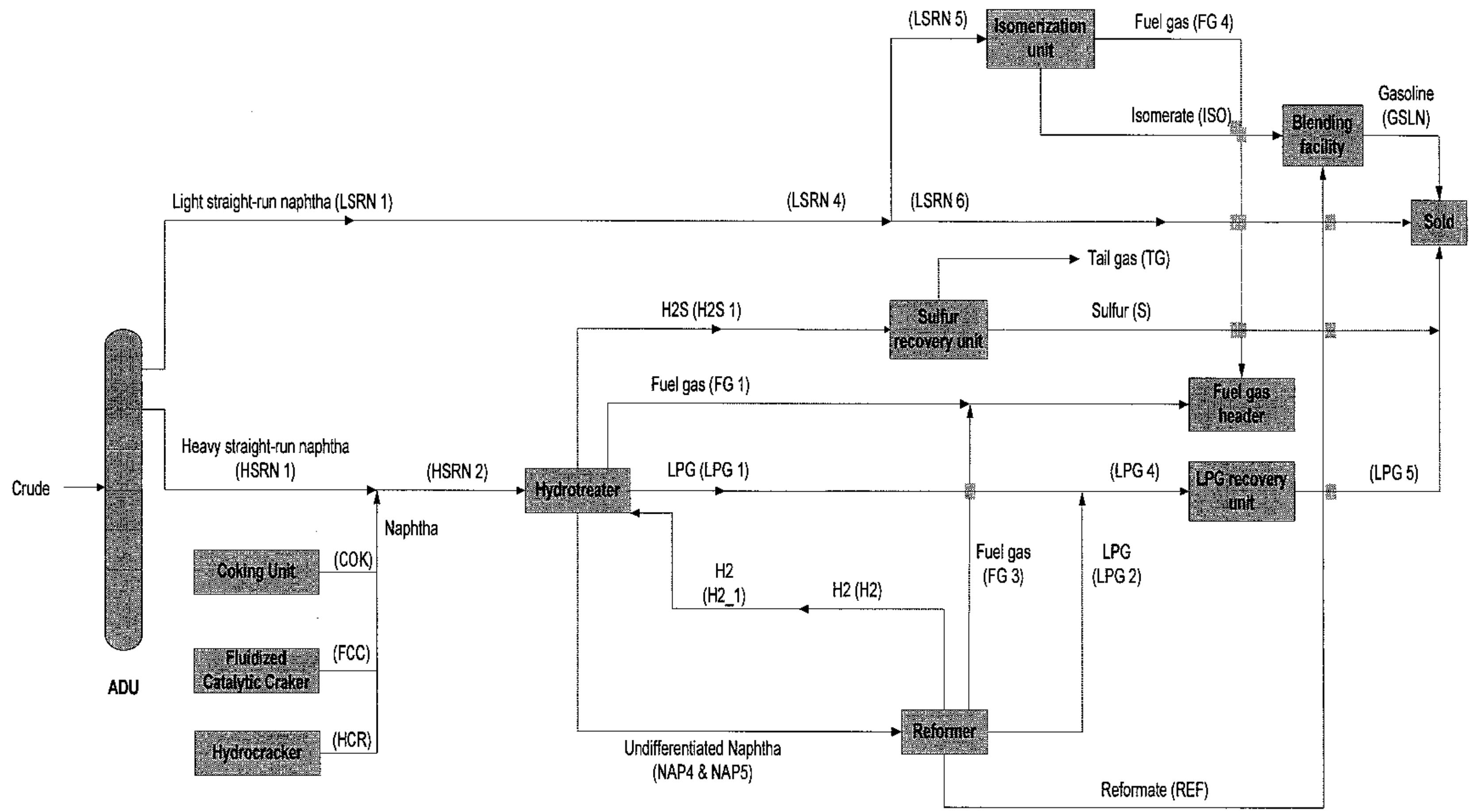


**Figure 4.4:** Optimal flowsheet for naphtha with heavy crude charge API $\leq$ 33° and the specified input data (MILP)

## 4.4 COMPUTATIONAL RESULTS DISCUSSION FOR MILP

OPEX for heavy crude charge processing is more than light crude charge processing for crude feed requirement and operating cost dependent MILP model. The comparisons for two cases are made for result validation on different scenarios. The result is proven right as heavy crude charge processing requires more severe processes, where COK, FCC and HCR are enforced to exist; while for light crude charge processing, VIS and FCC are enforced to exist. The operating cost for COK and HCR are relatively higher as compared to FCC and VIS, thus OPEX for heavy crude charge processing is higher. The results are tabulated in Table 4.5 and 4.8. The relative accuracy of objective function agrees with typical refineries cost data (e. g., estimated total cost of Bharat Petr Corp Ltd Refinery, Mahul, India is RM 2700 mil) stated by HPI Construction Boxscore, *Hydrocarbon Processing* (June 2006).

The optimal refinery topology for two scenarios is evaluated for MILP model, namely light crude charge processing and heavy crude charge processing. Figure 4.1 and Figure 4.2 show the superstructure comprising all the feasible alternative flow for naphtha produced from the atmospheric distillation unit for both scenarios respectively. The shaded square indicates the unit is selected and the shaded circle indicates the state is selected with flowrate. Figure 4.3 and Figure 4.4 are the simplified optimal refinery topology deriving from the previous two figures, similarly for both scenarios respectively. The optimal refinery topology for MILP model agrees with the typical existing topology of the refinery as shown in the literature review. (Hyperion Refining Energy Center, 2007)

Table 4.6 and Table 4.7 shows the continuous decision in which the optimum flowrate of each selected states for crude feed requirement dependent MILP model, for light and heavy crude charge processing respectively; while Table 4.9 and Table 4.10 shows the optimum flowrates for operating cost dependent MILP model.

## 4.5 COMPUTATIONAL RESULT FOR GDP

**Input data:**

- total feed flowrate from units in the refinery constant, non-dependent on light or heavy crude (feed from VIS, COK, HCR are set at 2000000 kg/d while from FCC set at 3000000 kg/d)
- production requirements are constant.

**Table 4.11:** Computational results on the cost components for GDP

| | **Light Crude** | **Heavy Crude** |
|---|---|---|
| CAPEX + OPEX + Raw material (*mil RM*) | 2746 | 2747 |
| CAPEX (*mil RM*) | 791 | 791 |
| OPEX + Raw material (*mil RM*) | 1955 | 1951 |
| Crude feed requirement (*kg/d*) | 4.5E+7 | 3.2E+7 |
| Raw material (*mil RM*) | 42 | 30 |
| OPEX (*mil RM*) | 1912 | 1920 |
| OPEX (*mil RM/d*) | 5.7 | 5.8 |

Note: assuming 330 working days

**Table 4.12:** Computational results on the stream flow rates for GDP (light crude)

| **Stream** | **Flowrate (kg/d)** | **Stream** | **Flow rate( kg/d)** | **Stream** | **Flow rate (kg/d)** |
|---|---|---|---|---|---|
| COK_1 | 0 | HCR_4 | 0 | NAP1 | 0 |
| COK_2 | 0 | HSRN1 | 6958014 | NAP2 | 2.3E+7 |
| FCC_1 | 2000000 | HSRN2 | 1.0E+7 | NAP3 | 0 |
| FCC_2 | 0 | HSRN3 | 4235584 | NAP4 | 0 |
| FG1 | 127554 | HSRN4 | 0 | NAP5 | 2.3E+7 |
| FG2 | 0 | HSRN5 | 0 | PCHN1_1 | 0 |
| FG3 | 849952 | ISO | 2244654 | PCHN1_2 | 0 |
| FG4 | 22765 | LPG1 | 67913 | PCHN2 | 0 |
| FG5 | 1000000 | LPG2 | 1791791 | PCHN3_1 | 0 |
| GSLN | 2.2E+7 | LPG3 | 0 | PCHN3_2 | 0 |
| H2 | 735106 | LPG4 | 1859516 | REF | 1.9E+7 |
| H2_1 | 735106 | LPG5 | 1859516 | S | 11987 |
| H2_2 | 0 | LSRN1 | 2519466 | SOLD | 2.4E+7 |
| H2S1 | 14123 | LSRN2 | 0 | TG | 2226 |
| H2S2 | 0 | LSRN3 | 0 | VIS_1 | 2000000 |
| HCR_1 | 0 | LSRN4 | 2519466 | VIS_2 | 0 |
| HCR_2 | 0 | LSRN5 | 2267528 | CR | 4.6E+7 |
| HCR_3 | 0 | LSRN6 | 251148 | | |

**Table 4.13:** Computational results on the stream flow rates for GDP (heavy crude)

| Stream | Flowrate (kg/d) | Stream | Flow rate (kg/d) | Stream | Flow rate (kg/d) |
|---|---|---|---|---|---|
| COK_1 | 2000000 | HCR_4 | 0 | NAP1 | 0 |
| COK_2 | 0 | HSRN1 | 5029521 | NAP2 | 0 |
| FCC_1 | 2000000 | HSRN2 | 1.1E+7 | NAP3 | 0 |
| FCC_2 | 0 | HSRN3 | 0 | NAP4 | 2.4E+7 |
| FG1 | 128194 | HSRN4 | 0 | NAP5 | 2.4E+7 |
| FG2 | 0 | HSRN5 | 0 | PCHN1_1 | 0 |
| FG3 | 855419 | ISO | 1622447 | PCHN1_2 | 0 |
| FG4 | 16478 | LPG1 | 68352 | PCHN2 | 0 |
| FG5 | 1000000 | LPG2 | 1803216 | PCHN3_1 | 0 |
| GSLN | 2.2E+7 | LPG3 | | PCHN3_2 | 0 |
| H2 | 739834 | LPG4 | 1871477 | REF | 1.9E+7 |
| H2_1 | 739834 | LPG5 | 1871477 | S | 11164 |
| H2_2 | 0 | LSRN1 | 1820918 | SOLD | 2.4E+7 |
| H2S1 | 14223 | LSRN2 | 0 | TG | 2231 |
| H2S2 | 0 | LSRN3 | 0 | VIS_1 | 0 |
| HCR_1 | 2000000 | LSRN4 | 1820918 | VIS_2 | 0 |
| HCR_2 | 0 | LSRN5 | 1638835 | CR | 3.3E+7 |
| HCR_3 | 0 | LSRN6 | 182173 | | 0 |

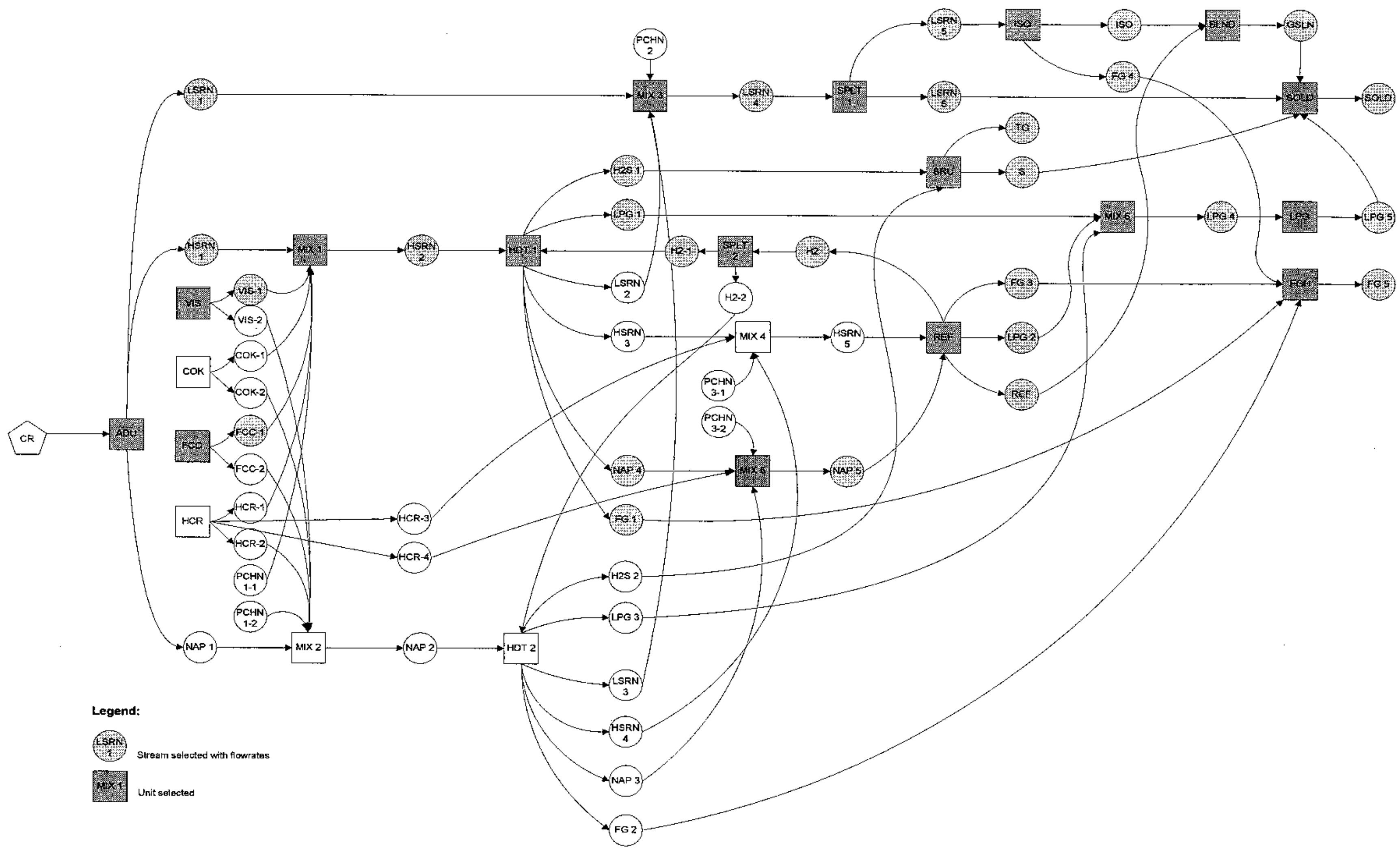


**Figure 4.5:** Optimal superstructure topology naphtha produced from ADU with light crude charge (GDP)

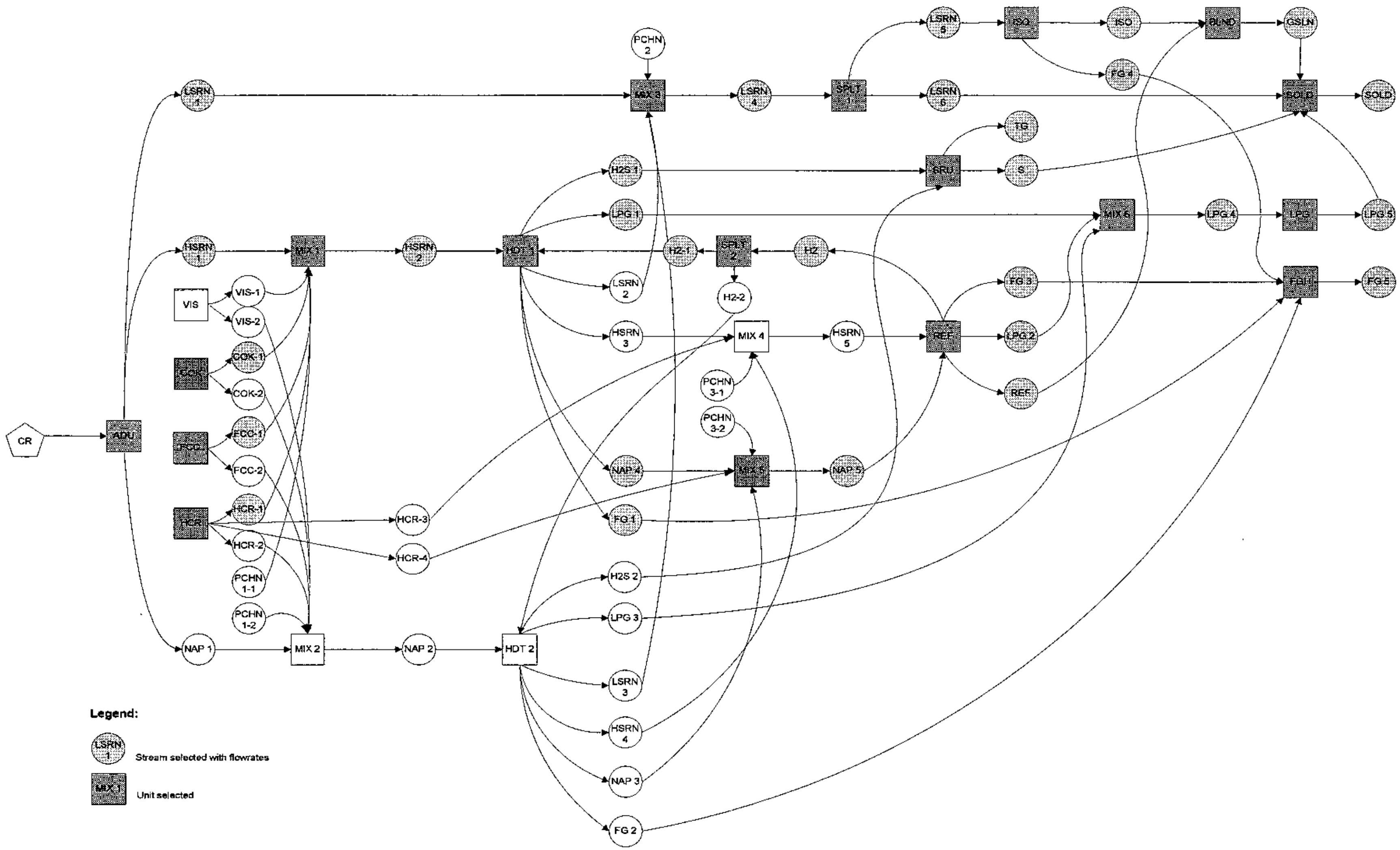


**Figure 4.6:** Optimal superstructure topology naphtha produced from ADU with heavy crude charge (GDP)

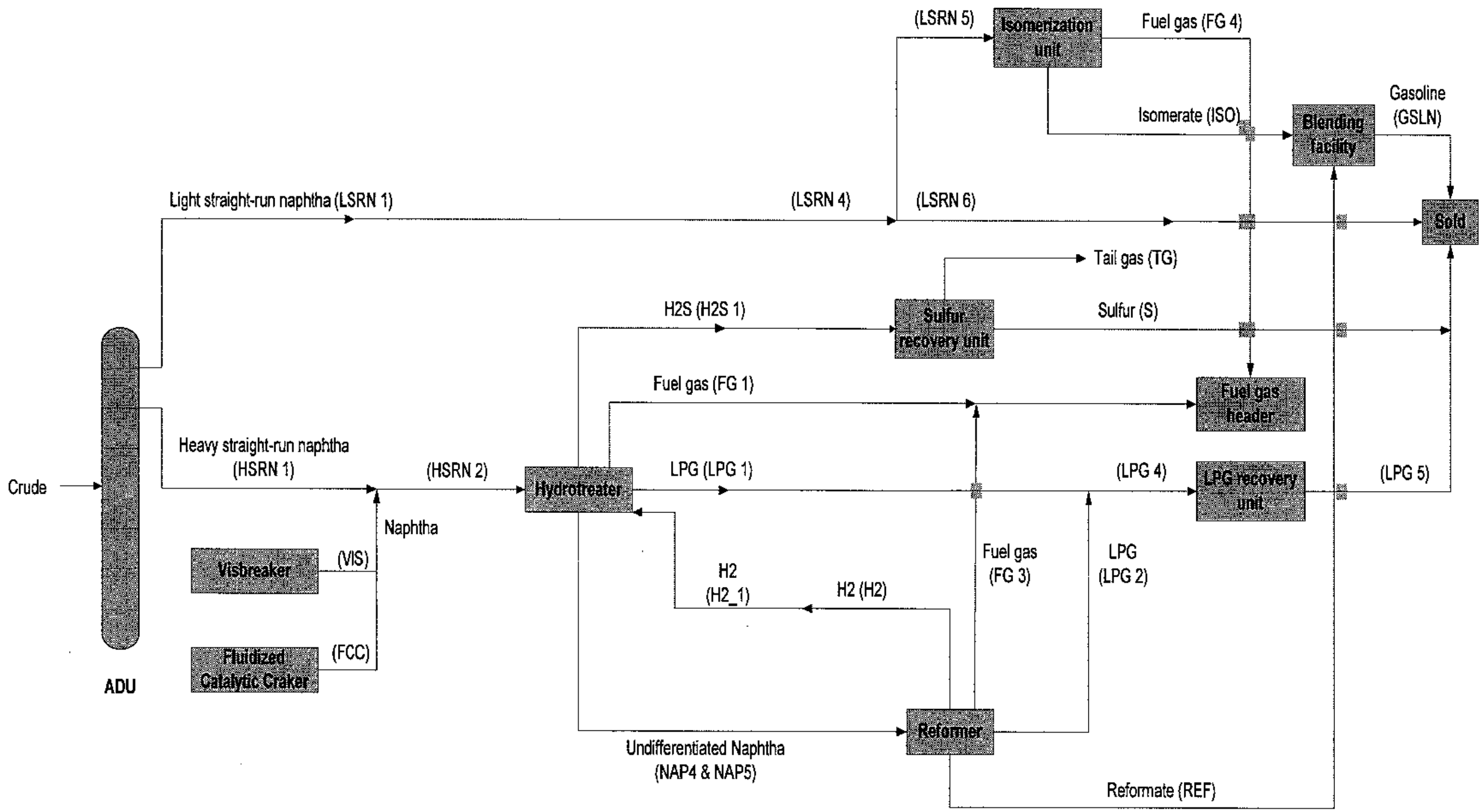


**Figure 4.7:** Optimal flowsheet for naphtha with light crude charge API > 33° and the specified input data (GDP)

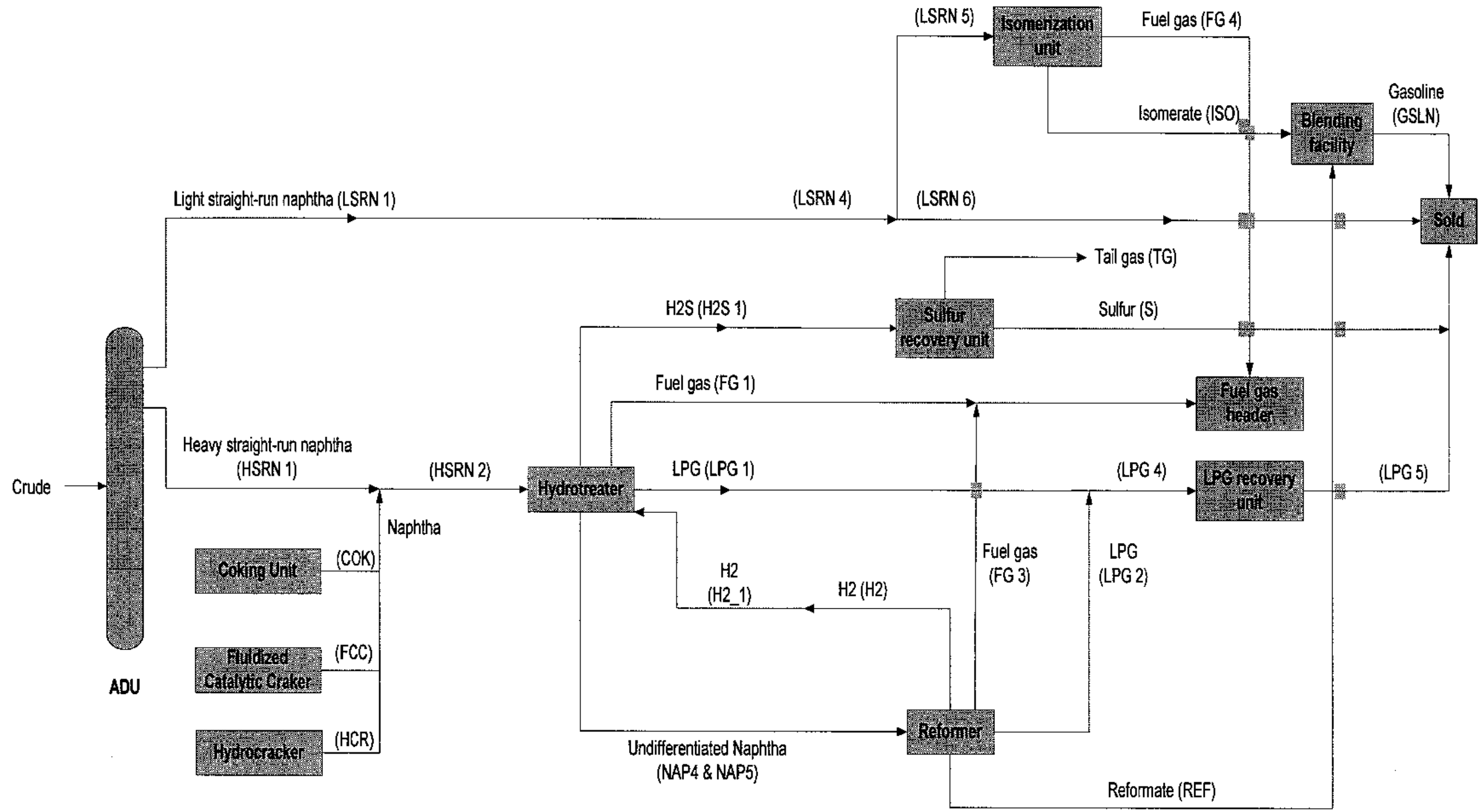


**Figure 4.8:** Optimal flowsheet for naphtha with heavy crude charge API $\leq 33°$ and the specified input data (GDP)

## 4.6 COMPUTATIONAL RESULTS DISCUSSION FOR GDP

Table 4.11 for GDP model agrees with the findings of MILP model that OPEX for heavy crude charge processing is more than light crude charge processing. The total cost (objective function) also as predicted, which agrees with typical refineries cost data (e. g., estimated total cost of Bharat Petr Corp Ltd Refinery, Mahul, India is RM 2700 mil) stated by HPI Construction Boxscore, *Hydrocarbon Processing* (June 2006).

Table 4.12 and Table 4.13 shows the continuous decision in which the optimum flowrate of each selected states for MILP model, for light and heavy crude charge processing respectively.

The optimal refinery topology for two scenarios is evaluated for GDP model, namely light crude charge processing and heavy crude charge processing. Figure 4.7 and Figure 4.8 shows the superstructure comprising all the feasible alternative flow for naphtha produced from the atmospheric distillation unit for both scenarios respectively. The shaded square indicates the unit is selected and the shaded circle indicates the state is selected with flowrate. Figure 4.3 and Figure 4.4 are the simplified optimal refinery topology deriving from the previous two figures, similarly for both scenarios respectively. The optimal refinery topology for MILP model agrees with the typical existing topology of the refinery as shown in the literature review. (Hyperion Refining Energy Center, 2007)

## 4.7 RESULT COMPARISON BETWEEN MILP AND GDP

The optimal refinery topology generated for MILP and GDP model agrees typical existing refinery topology. (Hydrocarbon Processing, 2006) The objective function of cost minimization is validated by comparing with the industrial data. (Hyperion Refining Energy Center, 2007) The only difference between the two model is that the CPU time/resource usage of GDP is lesser than MILP, in which GDP avoids the use of big-$M$ logical constraints, which yields poorer relaxation.

# CHAPTER 5
# CONCLUSIONS AND RECOMMENDATIONS

## 5.1 CONCLUSIONS

In this work, we have has successfully accomplished the following objectives that have been outlined. A superstructure representation embedding all feasible alternatives for naphtha produced from atmospheric distillation unit is developed with a suitable level of detail. Logical constraints on design and structural specifications for processing alternatives are developed. MILP and GDP are formulated for refinery design of naphtha processing alternatives. Solution of MILP is obtained using binary variables and big-M logical constraint; while solution of GDP is obtained using disjunction to determine optimal refinery subsystem topology/configuration for naphtha processing.

In conclusion, the optimal refinery topology obtained through MILP and GDP model agrees with typical existing topology of refineries (Gary and Handwerk, 1994) and the computational time are improved by using GDP formulation as compared to MILP formulation.

## 5.2 RECOMMENDATIONS FOR FUTURE WORK

The scope for future work includes:

1. To account for crude charges with an industrially-representative range of APIs;
2. To introduce nonlinearity in the model formulation to account for variable yields (compositions);
3. To incorporate more design specifications on important qualitative engineering heuristics, knowledge, and experience in refinery design via the use of logical constraints based on the reformulation of logic propositions.

# REFERENCES


CERA. 7 Nov 2007. <http://www.cera.com/aspx/cda/public1/news/pressReleases/>

Abuelsamid S. 17 Dec 2007. <http://www.autobloggreen.com/tag/hyperion-energy-center/>.

Ali, S. A. 2004. Naphtha hydrotreatment. In Antos, G. J. and Aitani, A. M. (editors). *Catalytic Naphtha Reforming*. New York : Marcel Dekker.

Barbosa-Povoa, A. P. F. D. and Pantelides, C. C. 1997. Design of Multipurpose Plants Using the Resource-Task Network Unified Framework. *Computers & Chemical Engineering, 21*, S703.

Biegler, L. T., Grossmann, I. E., and Westerberg, A. W. 1997. *Systematic Methods of Chemical Process Design*. New Jersey : Prentice Hall.

Clocksin W.F. and C.S.Mellish.1981. *Programming in Prolog*. Springer-Verlag, N. York.

Douglas, J.M. 1988.*Conceptual Design of Chemical Processes*.New York: McGraw-Hill.

Floudas, C. A. 1995. *Nonlinear and Mixed-Integer Optimization*. New York: Oxford University Press.

Gary, J. H. and Handwerk, G. E. 2001. *Petroleum Refining: Technology and Economics*. New York : Marcel Dekker .

Grossmann, I.E. 1996. *Mixed-integer optimization techniques for algorithmic process synthesis*. In J. L Anderson, Advances in chemical engineering vol. 23 (pp. 172-239). Academic Press.

HPI Construction Boxscore . *Hydrocarbon Processing* . June 2006

Maples, R. E. 2000. *Petroleum Refinery Process Economics*. Oklahoma: Pennwell.

Metin Turkay & Ignacio E. Grossmann.1995. Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Network. *Computers & Chemical Engineering*, 20: 960-978.

Metin Turkay and Ignacio E. Grossmann. 1996. Disjunctive Programming Techniques for the Optimization of Process System with Discontonous Investment Costs – Multiple Size Regions. Integration of logic and heuristic knowledge in MINLP optimization for process synthesis. *Ind. Eng. Chem. Res. Computers*, 35: 2611-2623.

Parkash, S. 2003. *Refining Processes Handbook*. Burlington, Massachusetts: Elsevier.

Prestvik, R., Moljord, K., Grande, K., and Holmen, A. Compositional analysis of naphtha and reformate. In Antos, G. J. and Aitani, A. M. (editors). *Catalytic Naphtha Reforming*. New York : Marcel Dekker.

Raman, R. and I. E. Grossmann. 1991. Relation between MILP modelling and logical inference for chemical process synthesis. *Computers & Chemical Engineering,15*, 73.

Raman, Ramesh and Ignacio E. Grossmann. 1993a. Symbolic integration of logic in mixed-integer linear programming techniques for process synthesis. *Computers & Chemical Engineering* 17, 9: 909–927.

Raman, Ramesh and Ignacio E. Grossmann. 1993b. Symbolic integration of logic in MILP branch and bound methods for the synthesis of process networks. *Annals of Operations Research* 42, 1, 169–191.

Raman, Ramesh and Ignacio E. Grossmann. 1994. Modelling and computational techniques for logic based integer programming. *Computers & Chemical Engineering* 18, 7: 563–578.

Rardin R. L. and U. Choe. 1979. Tighter relaxations of fixed charge network flow problems. *Goergia Institute of Technlogy, Industrial and Systems Engineering Report Series*. J-79-18, Atlanta.

Vecchietti A. and Grossmann I. E. 1998. LOGMIP: a disjunctive 0-1 non-linear optimizer for process system models. *Computers & Chemical Engineering*, 23, 555-565.

Yeomans, H. and Grossmann, I. E. 1999. A systematic modeling framework of superstructure optimization in process synthesis. *Computers & Chemical Engineering*, 23, 709.

# APPENDIX A

# MILP GAMS CODE

```
$TITLE: Superstructure Optimization for the Design of Petroleum Refineries: Alternatives for Naphtha Exiting ADU
$EOLCOM #

SETS

I     set of process units (tasks)
/
ADUu    atmospheric distillation unit
BLNDu   blending unit
COKu    coker
FCCu    catalytic cracker
FGHu    fuel gas hydrotreater
HCRu    hydrocraker
HDT1u   hydrotreater 1
HDT2u   hydrotreater 2
ISOu    isomerization unit
LPGu    LPG recovery unit
MIX1u   mixer 1
MIX2u   mixer 2
MIX3u   mixer 3
MIX4u   mixer 4
MIX5u   mixer 5
MIX6u   mixer 6
REFu    reformate
SPLT1u  spliter 1
SPLT2u  spliter 2
SRUu    sulfur recovery unit
SOLDu   sold unit
VISu    visbreaker
/

J     set of process streams (states)
/
BLNDs
COK_1s
COK_2s
CRs
FCCs
FCC_1s
FCC_2s
FG1s
FG2s
FG3s
FG4s
FG5s
GSLNs
H2s
H2_1s
H2_2s
H2S1s
H2S2s
HCR_1s
HCR_2s
HCR_3s
HCR_4s
HSRN1s
HSRN2s
HSRN3s
HSRN4s
HSRN5s
ISOs
LPG1s
LPG2s
LPG3s
LPG4s
LPG5s
LSRN1s
LSRN2s
LSRN3s
LSRN4s
LSRN5s
LSRN6s
NAP1s
NAP2s
NAP3s
NAP4s
NAP5s
PCHN1s
PCHN1_1s
PCHN1_2s
PCHN2s
PCHN3s
PCHN3_1s
PCHN3_2s
REFs
Ss
SOLDs
TGs
VISs
VIS_1s
VIS_2s
/

INLET_STREAM(I,J)
/
ADUu.(CRs)
BLNDu.(ISOs,REFs)
FGHu.(FG1s,FG2s,FG3s,FG4s)
HDT1u.(HSRN2s,H2_1s)
HDT2u.(NAP2s,H2_2s)
ISOu.(LSRN5s)
LPGu.(LPG4s)
MIX1u.(HSRN1s,VIS_1s,COK_1s,FCC_1s,HCR_1s,PCHN1_1s)
MIX2u.(NAP1s,VIS_2s,COK_2s,FCC_2s,HCR_2s,PCHN1_2s)
MIX3u.(LSRN1s,PCHN2s,LSRN2s,LSRN3s)
MIX4u.(HSRN3s,HSRN4s,PCHN3_1s,HCR_3s)
MIX5u.(NAP3s,NAP4s,PCHN3_2s,HCR_4s)
MIX6u.(LPG1s,LPG2s,LPG3s)
REFu.(HSRN5s,NAP5s)
SPLT1u.(LSRN4s)
SPLT2u.(H2s)
SRUu.(H2S1s,H2S2s)
SOLDu.(GSLNs,LSRN6s,Ss,LPG5s)
/

MIXER(I)
/
MIX1u
MIX2u
MIX3u
MIX4u
MIX5u
MIX6u
/

SPLITTER(I)
/
SPLT1u
SPLT2u
/

INLET_MIXER(MIXER,J)
/
MIX1u.(HSRN1s,VIS_1s,COK_1s,FCC_1s,HCR_1s,PCHN1_1s)
MIX2u.(NAP1s,VIS_2s,COK_2s,FCC_2s,HCR_2s,PCHN1_2s)
MIX3u.(LSRN1s,PCHN2s,LSRN2s,LSRN3s)
MIX4u.(HSRN3s,HSRN4s,PCHN3_1s)
MIX5u.(NAP3s,NAP4s,PCHN3_2s)
MIX6u.(LPG1s,LPG2s,LPG3s)
/

OUTLET_MIXER(MIXER,J)
/
MIX1u.HSRN2s
MIX2u.NAP2s
MIX3u.LSRN4s
MIX4u.HSRN5s
MIX5u.NAP5s
MIX6u.LPG4s
/;

ALIAS(J,J1);

PARAMETERS

M1(I)   upper bound or maximum capacity of process units
M2(J)   upper bound or maximum capacity of stream piping

CAPEX_M(MIXER)  capital cost of mixers
CAPEX_S(SPLITTER)  capital cost of splitters
;

M1(I) = 100000000;
M2(J) = 100000000;

CAPEX_M(MIXER) = 100;
CAPEX_S(SPLITTER) = 100;

SCALARS

cr_cst         crude oil cost (RM per bbl) /120/
cr_kg_per_bbl  crude oil amount (kg per bbl) /127.7/
pchn_cst       purchased naphtha cost (RM per kg) /0.524/
API            API gravity of crude charge /30/
;


POSITIVE VARIABLES
F(J)         stream flowrates
```

```
;

BINARY VARIABLES
Y(I),Z(J)
;

FREE VARIABLES
c     total cost of refinery
;

Equations
objfn          min total cost in (mil RM)


*--------------------------------------------------------------------------------------------
*MATERIAL BALANCES

mat_bal1
mat_bal2
mat_bal3
mat_bal4
mat_bal5
mat_bal6
mat_bal7
mat_bal8
mat_bal9
mat_bal10
mat_bal11
mat_bal12
mat_bal13
mat_bal14
mat_bal15
mat_bal16
mat_bal17
mat_bal18

MAT_BAL_MIXER

mat_bal20
mat_bal21
mat_bal22
mat_bal23
mat_bal24
mat_bal25
mat_bal26
mat_bal27
mat_bal28
mat_bal29

mat_bal30
mat_bal31
mat_bal32
mat_bal33
mat_bal34
mat_bal36
mat_bal35
mat_bal37
mat_bal38
mat_bal39

*--------------------------------------------------------------------------------------------
*PRODUCTION REQUIREMENTS

prodreq1
prodreq2
prodreq3
prodreq4
prodreq5
prodreq6
prodreq7


*--------------------------------------------------------------------------------------------
* COMPONENT FLOW RATES

yield1
yield2
yield3
yield4
yield5
yield6
yield7
yield8
yield9
yield10
yield11
yield12
yield13
yield14
yield15
yield16
yield17
yield18
yield19
yield20
yield21
yield22
yield23
yield24
yield25

*--------------------------------------------------------------------------------------------
* BIG-M LOGICAL CONSTRAINTS

BIG_M_LOGICON1
BIG_M_LOGICON2

*--------------------------------------------------------------------------------------------
*DESIGN LOGICAL CONSTRAINTS

LOGICON1
LOGICON2
LOGICON3
LOGICON4
LOGICON5
LOGICON6
LOGICON7
LOGICON8
LOGICON9
LOGICON10
LOGICON11
LOGICON12
LOGICON13
LOGICON14
LOGICON15
LOGICON16
LOGICON17
LOGICON18
LOGICON19
LOGICON20
LOGICON21
LOGICON22
LOGICON23
LOGICON24
LOGICON25
LOGICON26
LOGICON27
LOGICON28
LOGICON29
LOGICON30

*--------------------------------------------------------------------------------------------
*INTERCONNECTIVITY

*ADU
SS1
SS2
SS3
SS4
SS5
SS6

*HDT 1
SS7
SS8
SS9
SS10
SS11
SS12
SS13
SS14
SS15
SS16

*HDT 2
SS17
SS18
SS19
SS20
SS21
SS22
SS23
SS24
SS25
SS26

*ISO
SS27
SS28
SS29
SS30
SS31

*SRU
SS32
SS33
SS34
SS35
SS36
SS37

*REF
SS38
SS39
SS40
SS41
SS42
```

```
SS43
SS44
SS45

*SOLD
SS46
SS47
SS48
SS49
SS50
SS51
SS52

*BLND
SS53
SS54
SS55
SS56
SS57

*LPG
SS58
SS59
SS60
SS61

*FGH
SS62
SS63
SS64
SS65
SS66
SS67
SS68

*SPLT 1
SS69
SS70
SS71
SS72
SS73

*SPLT 2
SS74
SS75
SS76
SS77
SS78

*MIX 1
SS79
SS80
SS81
SS82
SS83
SS84
SS85
SS86
SS87

*MIX 2
SS88
SS89
SS90
SS91
SS92
SS93
SS94
SS95
SS96

*MIX 3
SS97
SS98
SS99
SS100
SS101
SS102
SS103

*MIX 4
SS104
SS105
SS106
SS107
SS108
SS109
SS110

*MIX 5
SS111
SS112
SS113
SS114
SS115
SS116
SS117

*MIX 6
SS118
SS119
SS120
SS121
SS122
SS123

*VIS
SS124
SS125
SS126

*COK
SS127
SS128
SS129

*FCC
SS130
SS131
SS132

*HCR
SS133
SS134
SS135
SS136
SS137
;


*-----------------------------------------------------------------------------------------------------------
*OBJECTIVE FUNCTION

OBJFN..

C =e=
*tci
0.45*2.4*(228*y('ADUu')+96*y('HDT1u')+ 96*y('HDT2u')+270*y('REFu')+42*y('ISOu')+30*y('SRUu'))
+
0.1*0.45*2.4*(228*y('ADUu')+96*y('HDT1u')+ 96*y('HDT2u')+270*y('REFu')+42*y('ISOu')+30*y('SRUu'))
+
*toc
*ge
0.3*(0.35*0.45*2.4*(228*y('ADUu')+96*y('HDT1u')+96*y('HDT2u')+270*y('REFu')+42*y('ISOu')+30*y('SRUu'))
+
*Electricity used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
(0.198*
(f('CRs')*0.0039+
(f('VIS_1s')+ f('VIS_2s'))*0.0039+
(f('COK_1s')+ f('COK_2s'))*0.0282+
(f('FCC_1s')+ f('FCC_2s'))*0.0078+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0.1402+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.1402+
(f('NAP5s')+f('HSRN5s'))*0.0078+
f('LSRN5s')*0.0078+
(f('H2S1s')+f('H2S2s'))*0.3132)

+
*Fuel used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.1081*
(f('CRs')*0.0826 +
(f('VIS_1s')+f('VIS_2s'))*0.066 +
```

```
(f('COK_1s')+f('COK_2s'))*0.0991 +
(f('FCC_1s')+ f('FCC_2s'))*0.0660 +
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0.2766+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.0248+
(f('NAP5s')+f('HSRN5s'))*0.2477+
f('LSRN5s')*0.0083+
(f('H2S1s')+f('H2S2s'))*0)

+
*HP Steam used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.005*
(f('CRs')*0.0888+
(f('VIS_1s')+f('VIS_2s'))*0.1776+
(f('COK_1s')+f('COK_2s'))*0.1421+
(f('FCC_1s')+ f('FCC_2s'))*0.071+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.0533+
(f('NAP5s')+f('HSRN5s'))*0.1421+
f('LSRN5s')*0.1279+
(f('H2S1s')+f('H2S2s'))*2.6636)

+
*CW used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.84*
(f('CRs')*0+
(f('VIS_1s')+f('VIS_2s'))*0+
(f('COK_1s')+f('COK_2s'))*0+
(f('FCC_1s')+ f('FCC_2s'))*0.0119+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0+
(f('NAP5s')+f('HSRN5s'))*0.003+
f('LSRN5s')*0+
(f('H2S1s')+f('H2S2s'))*0.1482))*330/1000000
+cr_cst*f('CRs')/cr_kg_per_bbl/1000000
+pchn_cst*( f('PCHN1_1s')+f('PCHN1_2s')+f('PCHN2s')+f('PCHN3_1s')+f('PCHN3_2s') ) )

+
*foc
0.35*0.45*2.4*(228*y('ADUu')+96*y('HDT1u')+96*y('HDT2u')+270*y('REFu')+42*y('ISOu')+30*y('SRUu'))

+
*voc
*Electricity used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
(0.198*
(f('CRs')*0.0039+
(f('VIS_1s')+ f('VIS_2s'))*0.0039+
(f('COK_1s')+ f('COK_2s'))*0.0282+
(f('FCC_1s')+ f('FCC_2s'))*0.0078+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0.1402+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.1402+
(f('NAP5s')+f('HSRN5s'))*0.0078+
f('LSRN5s')*0.0078+
(f('H2S1s')+f('H2S2s'))*0.3132)

+
*Fuel used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.1081*
(f('CRs')*0.0826 +
(f('VIS_1s')+f('VIS_2s'))*0.066 +
(f('COK_1s')+f('COK_2s'))*0.0991 +
(f('FCC_1s')+ f('FCC_2s'))*0.0660 +
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0.2766+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.0248+
(f('NAP5s')+f('HSRN5s'))*0.2477+
f('LSRN5s')*0.0083+
(f('H2S1s')+f('H2S2s'))*0)

+
*HP Steam used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.005*
(f('CRs')*0.0888+
(f('VIS_1s')+f('VIS_2s'))*0.1776+
(f('COK_1s')+f('COK_2s'))*0.1421+
(f('FCC_1s')+ f('FCC_2s'))*0.071+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.0533+
(f('NAP5s')+f('HSRN5s'))*0.1421+
f('LSRN5s')*0.1279+
(f('H2S1s')+f('H2S2s'))*2.6636)

+
*CW used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.84*
(f('CRs')*0+
(f('VIS_1s')+f('VIS_2s'))*0+
(f('COK_1s')+f('COK_2s'))*0+
(f('FCC_1s')+ f('FCC_2s'))*0.0119+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0+
(f('NAP5s')+f('HSRN5s'))*0.003+
f('LSRN5s')*0+
(f('H2S1s')+f('H2S2s'))*0.1482))*330/1000000
+cr_cst*f('CRs')/cr_kg_per_bbl/1000000
+pchn_cst*( f('PCHN1_1s')+f('PCHN1_2s')+f('PCHN2s')+f('PCHN3_1s')+f('PCHN3_2s'))

+ SUM(MIXER, CAPEX_M(MIXER)*Y(MIXER)) + SUM(SPLITTER, CAPEX_S(SPLITTER)*Y(SPLITTER))    # (capital cost for mixers and splitters)
+ SUM(J, 10*Z(J))    # (taken to be or assumed to be piping cost for the selected stream)
;
```

```
*------------------------------------------------------------------------------------------------------
*MATERIAL BALANCES

mat_bal1..   0.4176*f('CRs') =g= f('NAP1s')+f('LSRN1s')+f('HSRN1s');                                          # ADU
mat_bal2..   1.9821*(f('HSRN2s')+f('H2_1s')) =e= f('FG1s')+f('H2S1s')+f('LPG1s')+f('LSRN2s')+f('HSRN3s')+f('NAP4s');  # HDT1
mat_bal3..   1.9821*(f('NAP2s')+f('H2_2s')) =e= f('FG2s')+f('H2S2s')+f('LPG3s')+f('LSRN3s')+f('HSRN4s')+f('NAP3s');   # HDT2
mat_bal4..   f('ISOs')+f('FG4s') =e= f('LSRN5s');                                                              # ISO
mat_bal5..   f('Ss')+f('TGs') =e= f('H2S1s')+f('H2S2s');                                                       # SRU
mat_bal6..   f('HSRN5s')+f('NAP5s') =e= f('H2s')+f('FG3s')+f('LPG2s')+f('REFs');                               # REF
mat_bal7..   f('SOLDs') =e= f('LSRN6s')+f('Ss')+f('GSLNs')+f('LPG5s');                                         # SOLD
mat_bal8..   f('GSLNs') =e= f('ISOs')+f('REFs');                                                               # BLND
mat_bal9..   f('LPG5s') =e= f('LPG4s');                                                                        # LPG
mat_bal10..  f('FG5s') =e= f('FG1s')+f('FG2s')+f('FG3s')+f('FG4s');                                            # FGH
mat_bal11..  f('LSRN4s') =e= f('LSRN5s')+f('LSRN6s');                                                          # SPLT1
mat_bal12..  f('H2s') =e= f('H2_1s')+f('H2_2s');                                                               # SPLT2
mat_bal13..  f('HSRN1s')+f('VIS_1s')+f('COK_1s')+f('FCC_1s')+f('HCR_1s')+f('PCHN1_1s') =e= f('HSRN2s');       # MIX1
mat_bal14..  f('NAP1s')+f('VIS_2s')+f('COK_2s')+f('FCC_2s')+f('HCR_2s')+f('PCHN1_2s') =e= f('NAP2s');         # MIX2
mat_bal15..  f('LSRN1s')+f('LSRN2s')+f('LSRN3s')+f('PCHN2s') =e= f('LSRN4s');                                  # MIX3
mat_bal16..  f('HSRN3s')+f('HSRN4s')+f('PCHN3_1s')+f('HCR_3s') =e= f('HSRN5s');                                # MIX4
mat_bal17..  f('NAP3s')+f('NAP4s')+f('PCHN3_2s')+f('HCR_4s') =e= f('NAP5s');                                   # MIX5
mat_bal18..  f('LPG1s')+f('LPG2s')+f('LPG3s') =e= f('LPG4s');                                                  # MIX6

MAT_BAL_MIXER(MIXER,J)$OUTLET_MIXER(MIXER,J)..   SUM(J1$INLET_MIXER(MIXER,J1), F(J1)) =E= F(J);

*HEAVY CRUDE (API<33)
mat_bal20..  f('COK_1s')=l=2000000;
mat_bal21..  f('COK_2s')=l=2000000;
mat_bal22..  f('HCR_1s')=l=2000000;
mat_bal23..  f('HCR_2s')=l=2000000;
mat_bal24..  f('HCR_3s')=l=2000000;
mat_bal25..  f('HCR_4s')=l=2000000;
mat_bal26..  f('VIS_1s')=e=0;
mat_bal27..  f('VIS_2s')=e=0;
mat_bal28..  f('FCC_1s')=l=2000000;
mat_bal29..  f('FCC_2s')=l=2000000;

*LIGHT CRUDE (API>=33)
mat_bal30..  f('VIS_1s')=l=2000000;
mat_bal31..  f('VIS_2s')=l=2000000;
mat_bal32..  f('COK_1s')=e=0;
mat_bal33..  f('COK_2s')=e=0;
mat_bal34..  f('HCR_1s')=e=0;
mat_bal35..  f('HCR_2s')=e=0;
mat_bal36..  f('HCR_3s')=e=0;
mat_bal37..  f('HCR_4s')=e=0;
mat_bal38..  f('FCC_1s')=l=2000000;
mat_bal39..  f('FCC_2s')=l=2000000;


*------------------------------------------------------------------------------------------------------
*PRODUCTION REQUIREMENTS

prodreq1..   f('CRs') =g= 10000000;
prodreq2..   f('CRs') =l= 50000000;
prodreq3..   f('GSLNs')=g=7000000;
prodreq4..   f('LPG5s')=g=1000000;
prodreq5..   f('FG5s')=g=1000000;
prodreq6..   f('PCHN3_1s')=l=1000;
prodreq7..   f('PCHN3_2s')=l=1000;


*------------------------------------------------------------------------------------------------------
* COMPONENT FLOW RATES

yield1..   0.0555*f('CRs') =e= f('LSRN1s');                                        # ADU
yield2..   0.1533*f('CRs') =e= f('HSRN1s');                                        # ADU
yield3..   0.2088*f('CRs') =e= f('NAP1s');                                         # ADU

yield4..   0.0109*(f('HSRN2s')+f('H2_1s')) =e= f('FG1s');                          # HDT1
yield5..   0.0012*(f('HSRN2s')+f('H2_1s')) =e= f('H2S1s');                         # HDT1
yield6..   0.0058*(f('HSRN2s')+f('H2_1s')) =e= f('LPG1s');                         # HDT1
yield7..   0.9821*(f('HSRN2s')+f('H2_1s')) =e= f('LSRN2s')+f('HSRN3s')+f('NAP4s'); # HDT1
yield8..   2.763*f('LSRN2s')=e=f('HSRN3s');                                        # HDT1
yield9..   0.9821*(f('HSRN2s')+f('H2_1s')) =e= f('NAP4s');                         # HDT1

yield10..  0.0109*(f('NAP2s')+f('H2_2s')) =e= f('FG2s');                           # HDT2
yield11..  0.0012*(f('NAP2s')+f('H2_2s')) =e= f('H2S2s');                          # HDT2
yield12..  0.0058*(f('NAP2s')+f('H2_2s')) =e= f('LPG3s');                          # HDT2
yield13..  0.2610*(f('NAP2s')+f('H2_2s')) =e= f('LSRN3s');                         # HDT2
yield14..  0.7211*(f('NAP2s')+f('H2_2s')) =e= f('HSRN4s');                         # HDT2
yield15..  0.9821*(f('NAP2s')+f('H2_2s')) =e= f('NAP3s');                          # HDT2

yield16..  0.01*f('LSRN5s')=e=f('FG4s');                                           # ISO
yield17..  0.99*f('LSRN5s')=e=f('ISOs');                                           # ISO
yield18..  0.8478*(f('H2S1s')+f('H2S2s'))=e= f('Ss');                              # SRU
yield19..  0.1522*(f('H2S1s')+f('H2S2s'))=e= f('TGs');                             # SRU
yield20..  0.0320*(f('HSRN5s')+f('NAP5s'))=e= f('H2s');                            # REF
yield21..  0.0370*(f('HSRN5s')+f('NAP5s'))=e= f('FG3s');                           # REF
yield22..  0.0780*(f('HSRN5s')+f('NAP5s'))=e= f('LPG2s');                          # REF
yield23..  0.8530*(f('HSRN5s')+f('NAP5s'))=e= f('REFs');                           # REF
yield24..  0.9*f('LSRN4s') =e= f('LSRN5s');                                        # SPLT1
yield25..  0.1*f('LSRN4s')=e= f('LSRN6s');                                         # SPLT1

*------------------------------------------------------------------------------------------------------
* BIG-M LOGICAL CONSTRAINTS

BIG_M_LOGICON1(I,J)$INLET_STREAM(I,J)..   F(J) =L= M1(I)*Y(I);
```

```
BIG_M_LOGICON2(J)..            F(J) =L= M2(J)*Z(J);

*--------------------------------------------------------------------------------------------
*DESIGN LOGICAL CONSTRAINT

LOGICON1..     Y('ADUu')=E=1;
LOGICON2..     Z('LSRN1s')+Z('NAP1s')-Y('ADUu')=E=0;
LOGICON3..     Y('HDT1u')+Y('HDT2u')-Y('ADUu')=E=0;
LOGICON4..     Z('H2S1s')+Z('H2S2s')-Y('SRUu')=E=0;
LOGICON5..     Y('SRUu')=E=Y('ADUu');

LOGICON6..     Z('LSRN1s')+Z('LSRN3s')=E=Y('MIX3u');
LOGICON7..     Z('HSRN3s')+Z('HSRN4s')=E=Y('MIX4u');
LOGICON8..     Z('NAP3s')+Z('NAP4s')=E=Y('MIX5u');

LOGICON9..     2*Y('MIX3u')+ Y('MIX5u')=G=Y('HDT1u');
LOGICON10..    Y('MIX3u')+ Y('MIX4u')+ Y('MIX5u')=G=Y('HDT1u');
LOGICON11..    Y('MIX3u')+ 2*Y('MIX5u')=G=Y('HDT1u');
LOGICON12..    Y('MIX4u')+ 2*Y('MIX5u')=G=Y('HDT1u');

LOGICON13..    2*Y('MIX3u')+ Y('MIX5u')=G=Y('HDT2u');
LOGICON14..    Y('MIX3u')+ Y('MIX4u')+ Y('MIX5u')=G=Y('HDT2u');
LOGICON15..    Y('MIX3u')+ 2*Y('MIX5u')=G=Y('HDT2u');
LOGICON16..    Y('MIX4u')+ 2*Y('MIX5u')=G=Y('HDT2u');

LOGICON17..    Y('REFu')=E=Y('ADUu');
LOGICON18..    Z('HSRN5s')+Z('NAP5s')-Y('REFu')=E= 0;

LOGICON19..    Y('LPGu')=E=Y('ADUu');
LOGICON20..    Y('HDT1u')+Y('HDT2u')=E=Z('LPG1s')+Z('LPG2s')+Z('LPG3s')- Y('LPGu');

LOGICON21..    Y('FGHu')=G=Z('FG1s');
LOGICON22..    Y('FGHu')=G=Z('FG2s');
LOGICON23..    Y('FGHu')=G=Z('FG3s');
LOGICON24..    Y('FGHu')=G=Z('FG4s');
LOGICON25..    Z('FG1s')+ Z('FG2s')+ Z('FG3s')+Z('FG4s')=G=Y('FGHu');

LOGICON26..    Y('ISOu')=E=Y('HDT1u');

LOGICON27..    Y('BLNDu')=E=Z('ISOs');
LOGICON28..    Y('BLNDu')=G=Z('REFs');
LOGICON29..    Z('ISOs')+Z('REFs')=G=Y('BLNDu');

LOGICON30..    Y('SOLDu') =E= Y('ADUu');
;

*--------------------------------------------------------------------------------------------
*INTERCONNECTIVITY

*STRUCTURAL SPECIFICATIONS

*ADU
SS1..   Z('CRs')     =L= Y('ADUu');
SS2..   Y('ADUu')    =L= Z('CRs');
SS3..   Y('ADUu')    =L= Z('LSRN1s') + Z('HSRN1s') + Z('NAP1s');
SS4..   Z('LSRN1s')  =L= Y('ADUu');
SS5..   Z('HSRN1s')  =L= Y('ADUu');
SS6..   Z('NAP1s')   =L= Y('ADUu');

*HDT1
SS7..   Y('HDT1u')   =L= Z('HSRN2s') + Z('H2_1s');
SS8..   Z('HSRN2s')  =L= Y('HDT1u') ;
SS9..   Z('H2_1s')   =L= Y('HDT1u') ;
SS10..  Y('HDT1u')   =L= Z('H2S1s') + Z('LPG1s') + Z('LSRN2s') + Z('HSRN3s') + Z('NAP4s') + Z('FG1s');
SS11..  Z('H2S1s')   =L= Y('HDT1u');
SS12..  Z('LPG1s')   =L= Y('HDT1u');
SS13..  Z('LSRN2s')  =L= Y('HDT1u');
SS14..  Z('HSRN3s')  =L= Y('HDT1u');
SS15..  Z('NAP4s')   =L= Y('HDT1u');
SS16..  Z('FG1s')    =L= Y('HDT1u');

*HDT2
SS17..  Y('HDT2u')   =L= Z('NAP2s') + Z('H2_2s');
SS18..  Z('NAP2s')   =L= Y('HDT2u') ;
SS19..  Z('H2_2s')   =L= Y('HDT2u') ;
SS20..  Y('HDT2u')   =L= Z('H2S2s') + Z('LPG3s') + Z('LSRN3s') + Z('HSRN4s') + Z('NAP3s') + Z('FG2s');
SS21..  Z('H2S2s')   =L= Y('HDT2u');
SS22..  Z('LPG3s')   =L= Y('HDT2u');
SS23..  Z('LSRN3s')  =L= Y('HDT2u');
SS24..  Z('HSRN4s')  =L= Y('HDT2u');
SS25..  Z('NAP3s')   =L= Y('HDT2u');
SS26..  Z('FG2s')    =L= Y('HDT2u');

*ISO
SS27..  Z('LSRN5s')  =L= Y('ISOu');
SS28..  Y('ISOu')    =L= Z('LSRN5s');
SS29..  Y('ISOu')    =L= Z('ISOs') + Z('FG4s');
SS30..  Z('ISOs')    =L= Y('ISOu');
SS31..  Z('FG4s')    =E= Y('ISOu');

*SRU
SS32..  Y('SRUu')    =L= Z('H2S1s') + Z('H2S2s');
SS33..  Z('H2S1s')   =L= Y('SRUu');
SS34..  Z('H2S2s')   =L= Y('SRUu');
SS35..  Y('SRUu')    =L= Z('Ss') + Z('TGs');
SS36..  Z('Ss')      =L= Y('SRUu');
SS37..  Z('TGs')     =L= Y('SRUu');

*REF
```

```
SS38..  Y('REFu')   =L= Z('HSRN5s') + Z('NAP5s');
SS39..  Z('HSRN5s')  =L= Y('REFu');
SS40..  Z('NAP5s')   =L= Y('REFu');
SS41..  Y('REFu')    =L= Z('H2s') + Z('FG3s') + Z('LPG2s') + Z('REFs');
SS42..  Z('H2s')     =L= Y('REFu');
SS43..  Z('FG3s')    =L= Y('REFu');
SS44..  Z('LPG2s')   =L= Y('REFu');
SS45..  Z('REFs')    =L= Y('REFu');

*SOLD
SS46..  Y('SOLDu')   =L= Z('LSRN6s') + Z('GSLNs') + Z('Ss') + Z('LPG5s');
SS47..  Z('LSRN6s')  =L= Y('SOLDu');
SS48..  Z('GSLNs')   =L= Y('SOLDu');
SS49..  Z('Ss')      =L= Y('SOLDu');
SS50..  Z('LPG5s')   =L= Y('SOLDu');
SS51..  Y('SOLDu')   =L= Z('SOLDs');
SS52..  Z('SOLDs')   =L= Y('SOLDu');

*BLND
SS53..  Y('BLNDu')   =L= Z('ISOs') + Z('REFs');
SS54..  Z('ISOs')    =L= Y('BLNDu');
SS55..  Z('REFs')    =L= Y('BLNDu');
SS56..  Y('BLNDu')   =L= Z('GSLNs');
SS57..  Z('GSLNs')   =L= Y('BLNDu');

*LPG
SS58..  Z('LPG4s')   =L= Y('LPGu');
SS59..  Y('LPGu')    =L= Z('LPG4s');
SS60..  Y('LPGu')    =L= Z('LPG5s');
SS61..  Z('LPG5s')   =L= Y('LPGu');

*FGH
SS62..  Y('FGHu')    =L= Z('FG1s') + Z('FG2s') + Z('FG3s') + Z('FG4s');
SS63..  Z('FG1s')    =L= Y('FGHu');
SS64..  Z('FG2s')    =L= Y('FGHu');
SS65..  Z('FG3s')    =L= Y('FGHu');
SS66..  Z('FG4s')    =L= Y('FGHu');
SS67..  Y('FGHu')    =L= Z('FG5s');
SS68..  Z('FG5s')    =L= Y('FGHu');

*SPLT1
SS69..  Z('LSRN4s')  =L= Y('SPLT1u');
SS70..  Y('SPLT1u')  =L= Z('LSRN4s');
SS71..  Y('SPLT1u')  =L= Z('LSRN5s') + Z('LSRN6s');
SS72..  Z('LSRN5s')  =L= Y('SPLT1u');
SS73..  Z('LSRN6s')  =L= Y('SPLT1u');

*SPLT2
SS74..  Z('H2s')     =L= Y('SPLT2u');
SS75..  Y('SPLT2u')  =L= Z('H2s');
SS76..  Y('SPLT2u')  =L= Z('H2_1s') + Z('H2_2s');
SS77..  Z('H2_1s')   =L= Y('SPLT2u');
SS78..  Z('H2_2s')   =L= Y('SPLT2u');

*MIX1
SS79..  Y('MIX1u')   =L= Z('HSRN1s') + Z('VIS_1s') + Z('COK_1s') + Z('FCC_1s') + Z('HCR_1s') + Z('PCHN1_1s');
SS80..  Z('HSRN1s')  =L= Y('MIX1u');
SS81..  Z('VIS_1s')  =L= Y('MIX1u');
SS82..  Z('COK_1s')  =L= Y('MIX1u');
SS83..  Z('FCC_1s')  =L= Y('MIX1u');
SS84..  Z('HCR_1s')  =L= Y('MIX1u');
SS85..  Z('PCHN1_1s') =L= Y('MIX1u');
SS86..  Y('MIX1u')   =L= Z('HSRN2s');
SS87..  Z('HSRN2s')  =L= Y('MIX1u');

*MIX2
SS88..  Y('MIX2u')   =L= Z('NAP1s') + Z('VIS_2s') + Z('COK_2s') + Z('FCC_2s') + Z('HCR_2s') + Z('PCHN1_2s');
SS89..  Z('NAP1s')   =L= Y('MIX2u');
SS90..  Z('VIS_2s')  =L= Y('MIX2u');
SS91..  Z('COK_2s')  =L= Y('MIX2u');
SS92..  Z('FCC_2s')  =L= Y('MIX2u');
SS93..  Z('HCR_2s')  =L= Y('MIX2u');
SS94..  Z('PCHN1_2s') =L= Y('MIX2u');
SS95..  Y('MIX2u')   =L= Z('NAP2s');
SS96..  Z('NAP2s')   =L= Y('MIX2u');


*MIX3
SS97..  Y('MIX3u')   =L= Z('LSRN1s') + Z('PCHN2s') + Z('LSRN2s') + Z('LSRN3s');
SS98..  Z('LSRN1s')  =L= Y('MIX3u');
SS99..  Z('PCHN2s')  =L= Y('MIX3u');
SS100..  Z('LSRN2s')  =L= Y('MIX3u');
SS101..  Z('LSRN3s')  =L= Y('MIX3u');
SS102..  Y('MIX3u')   =L= Z('LSRN4s');
SS103..  Z('LSRN4s')  =L= Y('MIX3u');

*MIX4
SS104..  Y('MIX4u')   =L= Z('HSRN3s') + Z('HCR_3s') + Z('PCHN3_1s') + Z('HSRN4s');
SS105..  Z('HSRN3s')  =L= Y('MIX4u');
SS106..  Z('HCR_3s')  =L= Y('MIX4u');
SS107..  Z('PCHN3_1s') =L= Y('MIX4u');
SS108..  Z('HSRN4s')  =L= Y('MIX4u');
SS109..  Y('MIX4u')   =L= Z('HSRN5s');
SS110..  Z('HSRN5s')  =L= Y('MIX4u');

*MIX5
SS111..  Y('MIX5u')   =L= Z('NAP3s') + Z('NAP4s') + Z('HCR_4s') + Z('PCHN3_2s');
SS112..  Z('NAP3s')   =L= Y('MIX5u');
SS113..  Z('NAP4s')   =L= Y('MIX5u');
```

```
SS114..  Z('HCR_4s')  =L= Y('MIX5u');
SS115..  Z('PCHN3_2s') =L= Y('MIX5u');
SS116..  Y('MIX5u')   =L= Z('NAP5s');
SS117..  Z('NAP5s')   =L= Y('MIX5u');

*MIX6
SS118..  Y('MIX6u')   =L= Z('LPG1s') + Z('LPG2s') + Z('LPG3s');
SS119..  Z('LPG1s')   =L= Y('MIX6u');
SS120..  Z('LPG2s')   =L= Y('MIX6u');
SS121..  Z('LPG3s')   =L= Y('MIX6u');
SS122..  Y('MIX6u')   =L= Z('LPG4s');
SS123..  Z('LPG4s')   =L= Y('MIX6u');

*VIS
SS124..  Y('VISu')    =L= Z('VIS_1s') + Z('VIS_2s');
SS125..  Z('VIS_1s')  =L= Y('VISu');
SS126..  Z('VIS_2s')  =L= Y('VISu');

*COK
SS127..  Y('COKu')    =L= Z('COK_1s') + Z('COK_2s');
SS128..  Z('COK_1s')  =L= Y('COKu');
SS129..  Z('COK_2s')  =L= Y('COKu');

*FCC
SS130..  Y('FCCu')    =L= Z('FCC_1s') + Z('FCC_2s');
SS131..  Z('FCC_1s')  =L= Y('FCCu');
SS132..  Z('FCC_2s')  =L= Y('FCCu');

*HCR
SS133..  Y('HCRu')    =L= Z('HCR_1s') + Z('HCR_2s') + Z('HCR_3s') + Z('HCR_4s');
SS134..  Z('HCR_1s')  =L= Y('HCRu');
SS135..  Z('HCR_2s')  =L= Y('HCRu');
SS136..  Z('HCR_3s')  =L= Y('HCRu');
SS137..  Z('HCR_4s')  =L= Y('HCRu');


*------------------------------------------------------------------------------------------------
Model
naphtha_opt_hvy
/
objfn

*------------------------------------------------------------------------------------------------
*MATERIAL BALANCES

mat_bal1
mat_bal2
mat_bal3
mat_bal4
mat_bal5
mat_bal6
mat_bal7
mat_bal8
mat_bal9
mat_bal10
mat_bal11
mat_bal12
mat_bal13
mat_bal14
mat_bal15
mat_bal16
mat_bal17
mat_bal18

MAT_BAL_MIXER

mat_bal20
mat_bal21
mat_bal22
mat_bal23
mat_bal24
mat_bal25
mat_bal26
mat_bal27
mat_bal28
mat_bal29

*------------------------------------------------------------------------------------------------
*PRODUCTION REQUIREMENTS

prodreq3
prodreq4
prodreq5
prodreq6
prodreq7


*------------------------------------------------------------------------------------------------
* COMPONENT FLOW RATES

yield1
yield2
*yield3
yield4
yield5
yield6
*yield7
yield8
*yield9
yield10
yield11
yield12
yield13
yield14
yield15
yield16
yield17
yield18
yield19
yield20
yield21
yield22
yield23
yield24
yield25

*------------------------------------------------------------------------------------------------
* BIG-M LOGICAL CONSTRAINTS

BIG_M_LOGICON1
BIG_M_LOGICON2

*------------------------------------------------------------------------------------------------
*DESIGN LOGICAL CONSTRAINTS

*LOGICON1
LOGICON2
LOGICON3
LOGICON4
LOGICON5
LOGICON6
LOGICON7
LOGICON8
LOGICON9
LOGICON10
LOGICON11
LOGICON12
LOGICON13
LOGICON14
LOGICON15
LOGICON16
LOGICON17
LOGICON18
LOGICON19
LOGICON20
LOGICON21
LOGICON22
LOGICON23
LOGICON24
LOGICON25
LOGICON26
LOGICON27
LOGICON28
LOGICON29
LOGICON30
```

*--------------------------------------------------------------------------------------------------------
*INTERCONNECTIVITY
*ADU
SS1
SS2
SS3
SS4
SS5
SS6

*HDT1
SS7
SS8
SS9
SS10
SS11
SS12
SS13
SS14
SS15
SS16

*HDT2
SS17
SS18
SS19
SS20
SS21
SS22
SS23
SS24
SS25
SS26

*ISO
SS27
SS28
SS29
SS30
SS31

*SRU
SS32
SS33
SS34
SS35
SS36
SS37

*REF
SS38
SS39
SS40
SS41
SS42
SS43
SS44
SS45

*SOLD
SS46
SS47
SS48
SS49
SS50
SS51
SS52

*BLND
SS53
SS54
SS55
SS56
SS57

*LPG
SS58
SS59
SS60
SS61

*FGH
SS62
SS63
SS64
SS65
SS66
SS67
SS68

*SPLT1
SS69
SS70
SS71
SS72
SS73

*SPLT2
SS74
SS75
SS76
SS77
SS78

*MIX1
SS79
SS80
SS81
SS82
SS83
SS84
SS85
SS86
SS87

*MIX2
SS88
SS89
SS90
SS91
SS92
SS93
SS94
SS95
SS96

*MIX3
SS97
SS98
SS99
SS100
SS101
SS102
SS103

*MIX4
SS104
SS105
SS106
SS107
SS108
SS109
SS110

*MIX5
SS111
SS112
SS113
SS114
SS115
SS116
SS117

*MIX6
SS118
SS119
SS120
SS121
SS122
SS123

*VIS
SS124
SS125
SS126

*COK
SS127
SS128
SS129

*FCC
SS130
SS131
SS132

*HCR
SS133
SS134
SS135
SS136
SS137
/

*--------------------------------------------------------------------------------------------------------
naphtha_opt_lgt
/
objfn
*--------------------------------------------------------------------------------------------------------
*MATERIAL BALANCES

mat_bal1
mat_bal2
mat_bal3
mat_bal4
mat_bal5

mat_bal6
mat_bal7
mat_bal8
mat_bal9
mat_bal10
mat_bal11
mat_bal12
mat_bal13
mat_bal14
mat_bal15
mat_bal16
mat_bal17
mat_bal18

MAT_BAL_MIXER

mat_bal30
mat_bal31
mat_bal32
mat_bal33
mat_bal34
mat_bal35
mat_bal36
mat_bal37
mat_bal38
mat_bal39

*--------------------------------------------------------------------------------
*PRODUCTION REQUIREMENTS

prodreq3
prodreq4
prodreq5
prodreq6
prodreq7

*--------------------------------------------------------------------------------
* COMPONENT FLOW RATES

yield1
yield2
*yield3
yield4
yield5
yield6
*yield7
yield8
*yield9
yield10
yield11
yield12
yield13
yield14
yield15
yield16
yield17
yield18
yield19
yield20
yield21
yield22
yield23
yield24
yield25

*--------------------------------------------------------------------------------
* BIG-M LOGICAL CONSTRAINTS

BIG_M_LOGICON1
BIG_M_LOGICON2

*--------------------------------------------------------------------------------
*DESIGN LOGICAL CONSTRAINTS

*LOGICON1
LOGICON2
LOGICON3
LOGICON4
LOGICON5
LOGICON6
LOGICON7
LOGICON8
LOGICON9
LOGICON10
LOGICON11
LOGICON12
LOGICON13
LOGICON14
LOGICON15
LOGICON16
LOGICON17
LOGICON18
LOGICON19
LOGICON20
LOGICON21
LOGICON22
LOGICON23
LOGICON24
LOGICON25
LOGICON26
LOGICON27
LOGICON28
LOGICON29
LOGICON30

*--------------------------------------------------------------------------------
*INTERCONNECTIVITY

*ADU
SS1
SS2
SS3
SS4
SS5
SS6

*HDT1
SS7
SS8
SS9
SS10
SS11
SS12
SS13
SS14
SS15
SS16

*HDT2
SS17
SS18
SS19
SS20
SS21
SS22
SS23
SS24
SS25
SS26

*ISO
SS27
SS28
SS29
SS30
SS31

*SRU
SS32
SS33
SS34
SS35
SS36
SS37

*REF
SS38
SS39
SS40
SS41
SS42
SS43
SS44
SS45

*SOLD
SS46
SS47
SS48
SS49
SS50
SS51
SS52

*BLND
SS53
SS54
SS55
SS56
SS57

*LPG
SS58
SS59
SS60
SS61

*FGH
SS62
SS63
SS64
SS65
SS66
SS67
SS68

*SPLT1

```
SS69
SS70
SS71
SS72
SS73

*SPLT2
SS74
SS75
SS76
SS77
SS78

*MIX1
SS79
SS80
SS81
SS82
SS83
SS84
SS85
SS86
SS87

*MIX2
SS88
SS89
SS90
SS91
SS92
SS93
SS94
SS95
SS96

*MIX3
SS97
SS98
SS99
SS100
SS101
SS102
SS103

*MIX4
SS104
SS105
SS106
SS107
SS108
SS109
SS110

*MIX5
SS111
SS112
SS113
SS114
SS115
SS116
SS117

*MIX6
SS118
SS119
SS120
SS121
SS122
SS123

*VIS
SS124
SS125
SS126

*COK
SS127
SS128
SS129

*FCC
SS130
SS131
SS132

*HCR
SS133
SS134
SS135
SS136
SS137

/

;

OPTION LIMROW = 1000;
OPTION LIMCOL = 1000;

if((API lt 33),
Solve
naphtha_opt_hvy using mip minimizing c;

Display
c.l,f.l,y.l,z.l
;

else
Solve
naphtha_opt_lgt using mip minimizing c;
;

Display
c.l,f.l,y.l,z.l
;)
```

# APPENDIX B

# GDP GAMS CODE

```
$TITLE: Superstructure Optimization for the Design of Petroleum Refineries: Alternatives for Naphtha Exiting ADU
$EOLCOM #

SETS

I      set of process units (tasks)
/
ADUu    atmospheric distillation unit
BLNDu   blending unit
COKu    coker
FCCu    catalytic cracker
FGHu    fuel gas hydrotreater
HCRu    hydrocraker
HDT1u   hydrotreater 1
HDT2u   hydrotreater 2
ISOu    isomerization unit
LPGu    LPG recovery unit
MIX1u   mixer 1
MIX2u   mixer 2
MIX3u   mixer 3
MIX4u   mixer 4
MIX5u   mixer 5
MIX6u   mixer 6
REFu    reformate
SPLT1u  spliter 1
SPLT2u  spliter 2
SRUu    sulfur recovery unit
SOLDu   sold unit
VISu    visbreaker
/

J      set of process streams (states)
/
COK_1s
COK_2s
CRs
FCC_1s
FCC_2s
FG1s
FG2s
FG3s
FG4s
FG5s
GSLNs
H2s
H2_1s
H2_2s
H2S1s
H2S2s
HCR_1s
HCR_2s
HCR_3s
HCR_4s
HSRN1s
HSRN2s
HSRN3s
HSRN4s
HSRN5s
ISOs
LPG1s
LPG2s
LPG3s
LPG4s
LPG5s
LSRN1s
LSRN2s
LSRN3s
LSRN4s
LSRN5s
LSRN6s
NAP1s
NAP2s
NAP3s
NAP4s
NAP5s
PCHN1_1s
PCHN1_2s
PCHN2s
PCHN3_1s
PCHN3_2s
REFs
Ss
SOLDs
TGs
VIS_1s
VIS_2s
/

K      cost of process units (tasks)
/
ADUu    atmospheric distillation unit
BLNDu   blending unit
COKu    coker
FCCu    catalytic cracker
FGHu    fuel gas hydrotreater
HCRu    hydrocraker
HDT1u   hydrotreater 1
HDT2u   hydrotreater 2
ISOu    isomerization unit
LPGu    LPG recovery unit
MIX1u   mixer 1
MIX2u   mixer 2
MIX3u   mixer 3
MIX4u   mixer 4
MIX5u   mixer 5
MIX6u   mixer 6
REFu    reformate
SPLT1u  spliter 1
SPLT2u  spliter 2
SRUu    sulfur recovery unit
SOLDu   sold unit
VISu    visbreaker
/

MIXER(I)
/
MIX1u
MIX2u
MIX3u
MIX4u
MIX5u
MIX6u
/

SPLITTER(I)
/
SPLT1u
SPLT2u
/

INLET_MIXER(MIXER,J)
/
MIX1u.(HSRN1s,VIS_1s,COK_1s,FCC_1s,HCR_1s,PCHN1_1s)
MIX2u.(NAP1s,VIS_2s,COK_2s,FCC_2s,HCR_2s,PCHN1_2s)
MIX3u.(LSRN1s,PCHN2s,LSRN2s,LSRN3s)
MIX4u.(HSRN3s,HSRN4s,PCHN3_1s)
MIX5u.(NAP3s,NAP4s,PCHN3_2s)
MIX6u.(LPG1s,LPG2s,LPG3s)
/

OUTLET_MIXER(MIXER,J)
/
MIX1u.HSRN2s
MIX2u.NAP2s
MIX3u.LSRN4s
MIX4u.HSRN5s
MIX5u.NAP5s
MIX6u.LPG4s
/;

ALIAS(J,J1);

PARAMETERS
CAPEX_M(MIXER)  capital cost of mixers
CAPEX_S(SPLITTER)  capital cost of splitters;

CAPEX_M(MIXER) = 100;
CAPEX_S(SPLITTER) = 100;



SCALARS
cr_cst          crude oil cost (RM per bbl) /216/
cr_kg_per_bbl   crude oil amount (kg per bbl) /127.7/
pchn_cst        purchased naphtha cost (RM per kg) /0.524/
API             API gravity of crude charge /40/
;


POSITIVE VARIABLES
F(J)          stream flowrates
;

BINARY VARIABLES
Y(I),Z(J)
```

```
;

FREE VARIABLES
COST        total cost of refinery
c(K)        cost of equipment
;

Equations
objfn       min total cost in (mil RM)

*------------------------------------------------------------------------------------------
*MATERIAL BALANCES - Definitions of equations independent of discrete choices

mat_bal1
mat_bal2
mat_bal3
mat_bal4
mat_bal5
mat_bal6
mat_bal7
mat_bal8
mat_bal9
mat_bal10
mat_bal11
mat_bal12
mat_bal13
mat_bal14
mat_bal15
mat_bal16
mat_bal17
mat_bal18

DUMMY

*------------------------------------------------------------------------------------------
*DEFINITION OF DISJUNCTION'S EQUATIONS
*ADU
INOUT1_1,INOUT1_2,INOUT1_3,INOUT1_4,INOUT1_5,INOUT1_6,INOUT1_7
COST1_1

*HDT-1
INOUT2_1,INOUT2_2,INOUT2_3,INOUT2_4,INOUT2_5,INOUT2_6,INOUT2_7,I
NOUT2_8,INOUT2_9,INOUT2_10,INOUT2_11,INOUT2_12,INOUT2_13,INOUT2
_14
COST2_1

*HDT-2
INOUT3_1,INOUT3_2,INOUT3_3,INOUT3_4,INOUT3_5,INOUT3_6,INOUT3_7,I
NOUT3_8,INOUT3_9,INOUT3_10,INOUT3_11,INOUT3_12,INOUT3_13,INOUT3
_14
COST3_1

*ISO
INOUT4_1,INOUT4_2,INOUT4_3,INOUT4_4,INOUT4_5
COST4_1

*SRU
INOUT5_1,INOUT5_2,INOUT5_3,INOUT5_4,INOUT5_5,INOUT5_6
COST5_1

*REF
INOUT6_1,INOUT6_2,INOUT6_3,INOUT6_4,INOUT6_5,INOUT6_6,INOUT6_7,I
NOUT6_8,INOUT6_9,INOUT6_10
COST6_1

*SOLD
INOUT7_1,INOUT7_2,INOUT7_3,INOUT7_4,INOUT7_5 ,INOUT7_6
COST7_1

*BLND
INOUT8_1,INOUT8_2,INOUT8_3,INOUT8_4
COST8_1

*LPG
INOUT9_1,INOUT9_2,INOUT9_3
COST9_1

*FG
INOUT10_1,INOUT10_2,INOUT10_3,INOUT10_4,INOUT10_5,INOUT10_6
COST10_1

*SPLT 1
INOUT11_1,INOUT11_2,INOUT11_3,INOUT11_4,INOUT11_5
COST11_1

*SPLT 2
INOUT12_1,INOUT12_2,INOUT12_3,INOUT12_4
COST12_1

*MIX-1
INOUT13_1,INOUT13_2,INOUT13_3,INOUT13_4,INOUT13_5,INOUT13_6,INO
UT13_7,INOUT13_8
COST13_1

*MIX-2
INOUT14_1,INOUT14_2,INOUT14_3,INOUT14_4,INOUT14_5,INOUT14_6,INO
UT14_7,INOUT14_8
COST14_1

*MIX-3
INOUT15_1,INOUT15_2,INOUT15_3,INOUT15_4,INOUT15_5,INOUT15_6
COST15_1

*MIX-4
INOUT16_1,INOUT16_2,INOUT16_3,INOUT16_4,INOUT16_5,INOUT16_6
COST16_1

*MIX-5
INOUT17_1,INOUT17_2,INOUT17_3,INOUT17_4,INOUT17_5,INOUT17_6
COST17_1

*MIX-6
INOUT18_1,INOUT18_2,INOUT18_3,INOUT18_4,INOUT18_5
COST18_1
;
*------------------------------------------------------------------------------------------
*OBJECTIVE FUNCTION

OBJFN..

COST =e=
*tci
0.45*2.4*(228*y('ADUu')+144*y('VISu')+276*y('COKu')+515*y('FCCu')+569*y('HCRu')+96*y('HDT1u')+
96*y('HDT2u')+270*y('REFu')+42*y('ISOu')+30*y('SRUu'))
+
0.1*0.45*2.4*(228*y('ADUu')+144*y('VISu')+276*y('COKu')+515*y('FCCu')+569*y('HCRu')+96*y('HDT1u')+
96*y('HDT2u')+270*y('REFu')+42*y('ISOu')+30*y('SRUu'))
+
*toc
*ge
0.3*(0.35*0.45*2.4*(228*y('ADUu')+144*y('VISu')+276*y('COKu')+515*y('FCCu')+569*y('HCRu')+96*y('HDT1u')+
96*y('HDT2u')+270*y('REFu')+42*y('ISOu')+30*y('SRUu'))
+
*Electricity used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
(0.198*
(f('CRs')*0.0039+
(f('VIS_1s')+ f('VIS_2s'))*0.0039+
(f('COK_1s')+ f('COK_2s'))*0.0282+
(f('FCC_1s')+ f('FCC_2s'))*0.0078+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0.1402+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.1402+
(f('NAP5s')+f('HSRN5s'))*0.0078+
f('LSRN5s')*0.0078+
(f('H2S1s')+f('H2S2s'))*0.3132)

+
*Fuel used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.1081*
(f('CRs')*0.0826 +
(f('VIS_1s')+f('VIS_2s'))*0.066 +
(f('COK_1s')+f('COK_2s'))*0.0991 +
(f('FCC_1s')+ f('FCC_2s'))*0.0660 +
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0.2766+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.0248+
(f('NAP5s')+f('HSRN5s'))*0.2477+
```

```
f('LSRN5s')*0.0083+
(f('H2S1s')+f('H2S2s'))*0)

+
*HP Steam used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.005*
(f('CRs')*0.0888+
(f('VIS_1s')+f('VIS_2s'))*0.1776+
(f('COK_1s')+f('COK_2s'))*0.1421+
(f('FCC_1s')+ f('FCC_2s'))*0.071+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.0533+
(f('NAP5s')+f('HSRN5s'))*0.1421+
f('LSRN5s')*0.1279+
(f('H2S1s')+f('H2S2s'))*2.6636)

+
*CW used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.84*
(f('CRs')*0+
(f('VIS_1s')+f('VIS_2s'))*0+
(f('COK_1s')+f('COK_2s'))*0+
(f('FCC_1s')+ f('FCC_2s'))*0.0119+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0+
(f('NAP5s')+f('HSRN5s'))*0.003+
f('LSRN5s')*0+
(f('H2S1s')+f('H2S2s'))*0.1482))*330/1000000
+cr_cst*f('CRs')/cr_kg_per_bbl/1000000
+pchn_cst*( f('PCHN1_1s')+f('PCHN1_2s')+f('PCHN2s')+f('PCHN3_1s')+f('PCHN3_2s') ) )

+
*foc
0.35*0.45*2.4*(228*y('ADUu')+144*y('VISu')+276*y('COKu')+515*y('FCCu')+569*y('HCRu')+96*y('HDT1u')+
96*y('HDT2u')+270*y('REFu')+42*y('ISOu')+30*y('SRUu'))

+
*voc
*Electricity used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
(0.198*
(f('CRs')*0.0039+
(f('VIS_1s')+ f('VIS_2s'))*0.0039+
(f('COK_1s')+ f('COK_2s'))*0.0282+
(f('FCC_1s')+ f('FCC_2s'))*0.0078+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0.1402+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.1402+
(f('NAP5s')+f('HSRN5s'))*0.0078+
f('LSRN5s')*0.0078+
(f('H2S1s')+f('H2S2s'))*0.3132)

+
*Fuel used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.1081*
(f('CRs')*0.0826 +
(f('VIS_1s')+f('VIS_2s'))*0.066 +
(f('COK_1s')+f('COK_2s'))*0.0991 +
(f('FCC_1s')+ f('FCC_2s'))*0.0660 +
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0.2766+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.0248+
(f('NAP5s')+f('HSRN5s'))*0.2477+
f('LSRN5s')*0.0083+
(f('H2S1s')+f('H2S2s'))*0)

+
*HP Steam used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.005*
(f('CRs')*0.0888+
(f('VIS_1s')+f('VIS_2s'))*0.1776+
(f('COK_1s')+f('COK_2s'))*0.1421+
(f('FCC_1s')+ f('FCC_2s'))*0.071+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0.0533+
(f('NAP5s')+f('HSRN5s'))*0.1421+
f('LSRN5s')*0.1279+
(f('H2S1s')+f('H2S2s'))*2.6636)

+
*CW used for ADU, VIS, COK, FCC, HCR, HDT1, HDT2, REF, ISO, SRU
0.84*
(f('CRs')*0+
(f('VIS_1s')+f('VIS_2s'))*0+
(f('COK_1s')+f('COK_2s'))*0+
(f('FCC_1s')+ f('FCC_2s'))*0.0119+
(f('HCR_1s')+ f('HCR_2s')+ f('HCR_3s')+ f('HCR_4s'))*0+
(f('HSRN2s') +f('NAP2s') +f('H2_1s') +f('H2_2s'))*0+
(f('NAP5s')+f('HSRN5s'))*0.003+
f('LSRN5s')*0+
(f('H2S1s')+f('H2S2s'))*0.1482))*330/1000000
+cr_cst*f('CRs')/cr_kg_per_bbl/1000000
+pchn_cst*( f('PCHN1_1s')+f('PCHN1_2s')+f('PCHN2s')+f('PCHN3_1s')+f('PCHN3_2s'))

+ SUM(MIXER, CAPEX_M(MIXER)*Y(MIXER)) + SUM(SPLITTER, CAPEX_S(SPLITTER)*Y(SPLITTER))    # (capital cost for mixers and splitters)
+ (F('NAP2s') + F('NAP4s'))*0.00744    # (operating cost (utility cost) for operating streams; coefficient 0.0744 = 0.0248*0.3 is due to coefficient used in Loh (2008))
+ SUM(J, 10*Z(J))    # (taken to be or assumed to be piping cost for the selected stream)
;


*-----------------------------------------------------------------------------------------------------------------------
*MATERIAL BALANCES
```

```
mat_bal1..    0.4176*f('CRs') =e= f('NAP1s')+f('LSRN1s')+f('HSRN1s');
mat_bal2..    1.9821*(f('HSRN2s')+f('H2_1s')) =e= f('FG1s')+f('H2S1s')+f('LPG1s')+f('LSRN2s')+f('HSRN3s')+f('NAP4s');
mat_bal3..    1.9821*(f('NAP2s')+f('H2_2s')) =e= f('FG2s')+f('H2S2s')+f('LPG3s')+f('LSRN3s')+f('HSRN4s')+f('NAP3s');
mat_bal4..    f('ISOs')+f('FG4s') =e= f('LSRN5s');
mat_bal5..    f('Ss')+f('TGs') =e= f('H2S1s')+f('H2S2s');
mat_bal6..    f('HSRN5s')+f('NAP5s') =e= f('H2s')+f('FG3s')+f('LPG2s')+f('REFs');
mat_bal7..    f('SOLDs') =e= f('LSRN6s')+f('Ss')+f('GSLNs')+f('LPG5s');
mat_bal8..    f('GSLNs') =e= f('ISOs')+f('REFs');
mat_bal9..    f('LPG5s') =e= f('LPG4s');
mat_bal10..   f('FG5s') =e= f('FG1s')+f('FG2s')+f('FG3s')+f('FG4s');
mat_bal11..   f('LSRN4s') =e= f('LSRN5s')+f('LSRN6s');
mat_bal12..   f('H2s') =e= f('H2_1s')+f('H2_2s');
mat_bal13..   f('NAP1s')+f('VIS_2s')+f('COK_2s')+f('FCC_2s')+f('HCR_2s')+f('PCHN1_2s') =e= f('NAP2s');
mat_bal14..   f('HSRN1s')+f('VIS_1s')+f('COK_1s')+f('FCC_1s')+f('HCR_1s')+f('PCHN1_1s') =e= f('HSRN2s');
mat_bal15..   f('LSRN4s') =e= f('LSRN1s')+f('LSRN2s')+f('LSRN3s')+f('PCHN2s');
mat_bal16..   f('HSRN3s')+f('HSRN4s')+f('PCHN3_1s')+f('HCR_3s') =e= f('HSRN5s');
mat_bal17..   f('NAP3s')+f('NAP4s')+f('PCHN3_2s')+f('HCR_4s') =e= f('NAP5s');
mat_bal18..   f('LPG4s') =e= f('LPG1s')+f('LPG2s')+f('LPG3s');

DUMMY..      SUM (I,Y(I))=G=0;
*----------------------------------------------------------------------------------------------------------------------------------
*DEFINITION OF DISJUNCTION'S EQUATIONS

*ADU
INOUT1_1..    0.2088*f('CRs') =e= f('NAP1s');
INOUT1_2..    0.0555*f('CRs') =e= f('LSRN1s');
INOUT1_3..    0.1533*f('CRs') =e= f('HSRN1s');
INOUT1_4..    f('NAP1s') =e= 0;
INOUT1_5..    f('LSRN1s') =e= 0;
INOUT1_6..    f('HSRN1s') =e= 0;
INOUT1_7..    f('CRs') =e= 0;
COST1_1..     c('ADUu') =e= 228;
*COST1_2..    c('ADUu') =e= 0;

*HDT-1
INOUT2_1..    0.0109*(f('HSRN2s')+f('H2_1s')) =e= f('FG1s');
INOUT2_2..    0.0012*(f('HSRN2s')+f('H2_1s')) =e= f('H2S1s');
INOUT2_3..    0.0058*(f('HSRN2s')+f('H2_1s')) =e= f('LPG1s');
INOUT2_4..    0.2610*(f('HSRN2s')+f('H2_1s')) =e= f('LSRN2s');
INOUT2_5..    0.7211*(f('HSRN2s')+f('H2_1s')) =e= f('HSRN3s');
INOUT2_6..    0.9821*(f('HSRN2s')+f('H2_1s')) =e= f('NAP4s');
INOUT2_7..    f('FG1s') =e= 0;
INOUT2_8..    f('H2S1s') =e= 0;
INOUT2_9..    f('LPG1s') =e= 0;
INOUT2_10..   f('LSRN2s') =e= 0;
INOUT2_11..   f('HSRN3s') =e= 0;
INOUT2_12..   f('NAP4s') =e= 0;
INOUT2_13..   f('HSRN2s') =e= 0;
INOUT2_14..   f('H2_1s') =e= 0;
COST2_1..     c('HDT1u') =e= 96;
*COST2_2..    c('HDT1u') =e= 0;

*HDT-2
INOUT3_1..    0.0109*(f('NAP2s')+f('H2_2s')) =e= f('FG2s');
INOUT3_2..    0.0012*(f('NAP2s')+f('H2_2s')) =e= f('H2S2s');
INOUT3_3..    0.0058*(f('NAP2s')+f('H2_2s')) =e= f('LPG3s');
INOUT3_4..    0.2610*(f('NAP2s')+f('H2_2s')) =e= f('LSRN3s');
INOUT3_5..    0.7211*(f('NAP2s')+f('H2_2s')) =e= f('HSRN4s');
INOUT3_6..    0.9821*(f('NAP2s')+f('H2_2s')) =e= f('NAP3s');
INOUT3_7..    f('FG2s') =e= 0;
INOUT3_8..    f('H2S2s') =e= 0;
INOUT3_9..    f('LPG3s') =e= 0;
INOUT3_10..   f('LSRN3s') =e= 0;
INOUT3_11..   f('HSRN4s') =e= 0;
INOUT3_12..   f('NAP3s') =e= 0;
INOUT3_13..   f('NAP2s') =e= 0;
INOUT3_14..   f('H2_2s') =e= 0;
COST3_1..     c('HDT2u') =e= 96;
*COST3_2..    c('HDT2u') =e= 0;

*ISO
INOUT4_1..    0.99*f('LSRN5s') =e= f('ISOs');
INOUT4_2..    0.01*f('LSRN5s') =e= f('FG4s');
INOUT4_3..    f('ISOs') =e= 0;
INOUT4_4..    f('FG4s') =e= 0;
INOUT4_5..    f('LSRN5s') =e= 0;
COST4_1..     c('ISOu') =e= 42;
*COST4_2..    c('ISOu') =e= 0;

*SRU
INOUT5_1..    0.8478*(f('H2S1s')+f('H2S2s')) =e= f('Ss');
INOUT5_2..    0.1522*(f('H2S1s')+f('H2S2s')) =e= f('TGs');
INOUT5_3..    f('Ss') =e= 0;
INOUT5_4..    f('TGs') =e= 0;
INOUT5_5..    f('H2S1s') =e= 0;
INOUT5_6..    f('H2S2s') =e= 0;
COST5_1..     c('SRUu') =e= 30;
*COST5_2..    c('SRUu') =e= 0;

*REF
INOUT6_1..    0.032*(f('HSRN5s')+f('NAP5s')) =e= f('H2s');
INOUT6_2..    0.037*(f('HSRN5s')+f('NAP5s')) =e= f('FG3s');
INOUT6_3..    0.078*(f('HSRN5s')+f('NAP5s')) =e= f('LPG2s');
INOUT6_4..    0.853*(f('HSRN5s')+f('NAP5s')) =e= f('REFs');
INOUT6_5..    f('H2s') =e= 0;
INOUT6_6..    f('FG3s') =e= 0;
INOUT6_7..    f('LPG2s') =e= 0;
INOUT6_8..    f('REFs') =e= 0;
```

```
INOUT6_9..     f('HSRN5s') =e= 0;
INOUT6_10..    f('NAP5s') =e= 0;
COST6_1..      c('REFu') =e= 270;
*COST6_2..     c('REFu') =e= 0;

*SOLD
INOUT7_1..     f('SOLDs') =e= f('LSRN6s')+f('Ss')+f('GSLNs')+f('LPG5s');
INOUT7_2..     f('SOLDs') =e= 0;
INOUT7_3..     f('LSRN6s') =e= 0;
INOUT7_4..     f('Ss') =e= 0;
INOUT7_5..     f('GSLNs') =e= 0;
INOUT7_6..     f('LPG5s') =e= 0;
COST7_1..      c('SOLDu') =e= 10;
*COST7_2..     c('SOLDu') =e= 0;

*BLND
INOUT8_1..     f('GSLNs') =e= f('ISOs')+f('REFs');
INOUT8_2..     f('GSLNs') =e= 0;
INOUT8_3..     f('ISOs') =e= 0;
INOUT8_4..     f('REFs') =e= 0;
COST8_1..      c('BLNDu') =e= 10;
*COST8_2..     c('BLNDu') =e= 0;

*LPG
INOUT9_1..     f('LPG5s') =e= f('LPG4s');
INOUT9_2..     f('LPG5s') =e= 0;
INOUT9_3..     f('LPG4s') =e= 0;
COST9_1..      c('LPGu') =e= 10;
*COST9_2..     c('LPGu') =e= 0;

*FGH
INOUT10_1..    f('FG5s') =e= f('FG1s')+f('FG2s')+f('FG3s')+f('FG4s');
INOUT10_2..    f('FG5s') =e= 0;
INOUT10_3..    f('FG1s') =e= 0;
INOUT10_4..    f('FG2s') =e= 0;
INOUT10_5..    f('FG3s') =e= 0;
INOUT10_6..    f('FG4s') =e= 0;
COST10_1..     c('FGHu') =e= 10;
*COST10_2..    c('FGHu') =e= 0;

*SPLT 1
INOUT11_1..    0.9*f('LSRN4s') =e= f('LSRN5s');
INOUT11_2..    0.1*f('LSRN4s') =e= f('LSRN6s');
INOUT11_3..    f('LSRN4s') =e= 0;
INOUT11_4..    f('LSRN5s') =e= 0;
INOUT11_5..    f('LSRN6s') =e= 0;
COST11_1..     c('SPLT1u') =e= 10;
*COST11_2..    c('SPLT1u') =e= 0;

*SPLT 2
INOUT12_1..    f('H2s') =e= f('H2_1s')+f('H2_2s');
INOUT12_2..    f('H2s') =e= 0;
INOUT12_3..    f('H2_1s') =e= 0;
INOUT12_4..    f('H2_1s') =e= 0;
COST12_1..     c('SPLT2u') =e= 10;
*COST12_2..    c('SPLT2u') =e= 0;

*MIX-1
INOUT13_1..    f('NAP1s')+f('VIS_2s')+f('COK_2s')+f('FCC_2s')+f('HCR_2s')+f('PCHN1_2s') =e= f('NAP2s');
INOUT13_2..    f('NAP1s') =e= 0;
INOUT13_3..    f('VIS_2s') =e= 0;
INOUT13_4..    f('COK_2s') =e= 0;
INOUT13_5..    f('FCC_2s') =e= 0;
INOUT13_6..    f('HCR_2s') =e= 0;
INOUT13_7..    f('PCHN1_2s') =e= 0;
INOUT13_8..    f('NAP2s') =e= 0;
COST13_1..     c('MIX1u') =e= 10;
*COST13_2..    c('MIX1u') =e= 0;

*MIX-2
INOUT14_1..    f('HSRN1s')+f('VIS_1s')+f('COK_1s')+f('FCC_1s')+f('HCR_1s')+f('PCHN1_1s') =e= f('HSRN2s');
INOUT14_2..    f('HSRN1s') =e= 0;
INOUT14_3..    f('VIS_1s') =e= 0;
INOUT14_4..    f('COK_1s') =e= 0;
INOUT14_5..    f('FCC_1s') =e= 0;
INOUT14_6..    f('HCR_1s') =e= 0;
INOUT14_7..    f('PCHN1_1s') =e= 0;
INOUT14_8..    f('HSRN2s') =e= 0;
COST14_1..     c('MIX2u') =e= 10;
*COST14_2..    c('MIX2u') =e= 0;

*MIX-3
INOUT15_1..    f('LSRN4s') =e= f('LSRN1s')+f('LSRN2s')+f('LSRN3s')+f('PCHN2s');
INOUT15_2..    f('LSRN4s') =e= 0;
INOUT15_3..    f('LSRN1s') =e= 0;
INOUT15_4..    f('LSRN2s') =e= 0;
INOUT15_5..    f('LSRN3s') =e= 0;
INOUT15_6..    f('PCHN2s') =e= 0;
COST15_1..     c('MIX3u') =e= 10;
*COST15_2..    c('MIX3u') =e= 0;

*MIX-4
INOUT16_1..    f('HSRN3s')+f('HSRN4s')+f('PCHN3_1s')+f('HCR_3s') =e= f('HSRN5s');
INOUT16_2..    f('HSRN3s') =e= 0;
INOUT16_3..    f('HSRN4s') =e= 0;
INOUT16_4..    f('PCHN3_1s') =e= 0;
INOUT16_5..    f('HCR_3s') =e= 0;
INOUT16_6..    f('HSRN5s') =e= 0;
COST16_1..     c('MIX4u') =e= 10;
```

```
*COST16_2..    c('MIX4u') =e= 0;

*MIX-5
INOUT17_1..    f('NAP3s')+f('NAP4s')+f('PCHN3_2s')+f('HCR_4s') =e= f('NAP5s');
INOUT17_2..    f('NAP3s') =e= 0;
INOUT17_3..    f('NAP4s') =e= 0;
INOUT17_4..    f('PCHN3_2s') =e= 0;
INOUT17_5..    f('HCR_4s') =e= 0;
INOUT17_6..    f('NAP5s') =e= 0;
COST17_1..    c('MIX5u') =e= 10;
*COST17_2..    c('MIX5u') =e= 0;

*MIX-6
INOUT18_1..    f('LPG4s') =e= f('LPG1s')+f('LPG2s')+f('LPG3s');
INOUT18_2..    f('LPG4s') =e= 0;
INOUT18_3..    f('LPG1s') =e= 0;
INOUT18_4..    f('LPG2s') =e= 0;
INOUT18_5..    f('LPG3s') =e= 0;
COST18_1..    c('MIX5u') =e= 10;
*COST18_2..    c('MIX5u') =e= 0;

*------------------------------------------------------------------------------------------------------------------------------------------------
*BEGIN DECLARATIONS AND DEFINITIONS OF DISJUNCTIONS (LOGMIP SECTION)

$ONECHO>"%lm.info%"

Disjunction D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,D16,D17,D18;

D1 is if Y('ADUu') then
      INOUT1_1;
      INOUT1_2;
      INOUT1_3;
      COST1_1;
      else
      INOUT1_4;
      INOUT1_5;
      INOUT1_6;
      INOUT1_7;
      endif;

D2 is if Y('HDT-1u') then
      INOUT2_1;
      INOUT2_2;
      INOUT2_3;
      INOUT2_4;
      INOUT2_5;
      INOUT2_6;
      COST2_1;
      else
      INOUT2_7;
      INOUT2_8;
      INOUT2_9;
      INOUT2_10;
      INOUT2_11;
      INOUT2_12;
      INOUT2_13;
      INOUT2_14;
      endif;

D3 is if Y('HDT-2u') then
      INOUT3_1;
      INOUT3_2;
      INOUT3_3;
      INOUT3_4;
      INOUT3_5;
      INOUT3_6;
      COST3_1;
      else
      INOUT3_7;
      INOUT3_8;
      INOUT3_9;
      INOUT3_10;
      INOUT3_11;
      INOUT3_12;
      INOUT3_13;
      INOUT3_14;
      endif;

D4 is if Y('ISOu') then
      INOUT4_1;
      INOUT4_2;
      COST4_1;
      else
      INOUT4_3;
      INOUT4_4;
      INOUT4_5;
      endif;

D5 is if Y('SRUu') then
      INOUT5_1;
      INOUT5_2;
      COST4_1;
      else
      INOUT5_3;
      INOUT5_4;
      INOUT5_5;
      INOUT5_6;
      endif;

D6 is if Y('REFu') then
      INOUT6_1;
      INOUT6_2;
      INOUT6_3;
      INOUT6_4;
      COST6_1;
      else
      INOUT6_5;
      INOUT6_6;
      INOUT6_7;
      INOUT6_8;
      INOUT6_9;
      INOUT6_10;
      endif;

D7 is if Y('SOLDu') then
      INOUT7_1;
      COST7_1;
      else
      INOUT7_2;
      INOUT7_3;
      INOUT7_4;
      INOUT7_5;
      INOUT7_6;
      endif;

D8 is if Y('BLNDu') then
      INOUT8_1;
      COST8_1;
      else
      INOUT8_2;
      INOUT8_3;
      INOUT8_4;
      endif;

D9 is if Y('LPGu') then
      INOUT9_1;
      COST9_1;
      else
      INOUT9_2;
      INOUT9_3;
      endif;

D10 is if Y('FGHu') then
      INOUT10_1;
      COST10_1;
      else
      INOUT10_2;
      INOUT10_3;
      INOUT10_4;
      INOUT10_5;
      INOUT10_6;
      endif;

D11 is if Y('SPLT1u') then
      INOUT11_1;
      INOUT11_2;
      COST11_1;
      else
      INOUT11_3;
      INOUT11_4;
      INOUT11_5;
      endif;

D12 is if Y('SPLT2u') then
      INOUT12_1;
      COST12_1;
      else
      INOUT12_2;
      INOUT12_3;
      INOUT12_4;
```

```
      endif;

D13 is if Y('MIX1u') then
      INOUT13_1;
      COST13_1;
      else
      INOUT13_2;
      INOUT13_3;
      INOUT13_4;
      INOUT13_5;
      INOUT13_6;
      INOUT13_7;
      INOUT13_8;
      endif;

D14 is if Y('MIX2u') then
      INOUT14_1;
      COST14_1;
      else
      INOUT14_2;
      INOUT14_3;
      INOUT14_4;
      INOUT14_5;
      INOUT14_6;
      INOUT14_7;
      INOUT14_8;
      endif;

D15 is if Y('MIX3u') then
      INOUT15_1;
      COST15_1;
      else
      INOUT15_2;
      INOUT15_3;
      INOUT15_4;
      INOUT15_5;
      INOUT15_6;
      endif;

D16 is if Y('MIX4u') then
      INOUT16_1;
      COST16_1;
      else
      INOUT16_2;
      INOUT16_3;
      INOUT16_4;
      INOUT16_5;
      INOUT16_6;
      endif;

D17 is if Y('MIX5u') then
      INOUT17_1;
      COST17_1;
      else
      INOUT17_2;
      INOUT17_3;
      INOUT17_4;
      INOUT17_5;
      INOUT17_6;
      endif;

D18 is if Y('MIX6u') then
      INOUT18_1;
      COST18_1;
      else
      INOUT18_2;
      INOUT18_3;
      INOUT18_4;
      INOUT18_5;
      endif;

*DESIGN SPECIFICATIONS

ATMOST(Y('HDT1u'),Y('HDT2u'));
ATMOST(Z('LSRN1s'),Z('LSRN3s'));
ATMOST(Z('HSRN3s'),Z('HSRN4s'));
ATMOST(Z('NAP3s'),Z('NAP3s'));

Y('ADUu')      <-> Y('HDT1u') or Y('HDT2u');
Y('ISOu')      <-> Y('SPLT1u');
Y('SRUu')      <-> Y('HDT1u') or Y('HDT2u');
Y('REFu')      <-> Y('MIX4u') or Y('MIX5u');
Y('SOLDu')     <-> Y('HDT1u') or Y('HDT2u');
Y('BLNDu')     <-> Y('ISOu') and Y('REFu');
Y('LPGu')      <-> Y('MIX6u');
Y('FGHu')      <-> Y('HDT1u') or Y('HDT2u');
Y('SPLT1u')    <-> Y('MIX3u');
Y('SPLT2u')    <-> Y('REFu');
Y('MIX1u')     <-> Y('HDT1u');
Y('MIX2u')     <-> Y('HDT2u');
Y('MIX3u')     <-> Z('LSRN1s') or Z('LSRN3s');
Y('MIX4u')     <-> Z('HSRN3s') or Z('HSRN4s');
Y('MIX5u')     <-> Z('NAP3s') or Z('NAP4s');
Y('MIX6u')     <-> Y('HDT1u') or Y('HDT2u');

*STRUCTURAL SPECIFICATIONS

*ADU
Z('CRs') -> Y('ADUu');
Y('ADUu') -> Z('CRs');
Y('ADUu') -> Z('LSRN1s') or Z('HSRN1s') or Z('NAP1s');
Z('LSRN1s') -> Y('ADUu');
Z('HSRN1s') -> Y('ADUu');
Z('NAP1s') -> Y('ADUu');

*HDT 1
Y('HDT1u') -> Z('HSRN2s') or Z('HS2_1s');
Z('HSRN2s') -> Y('HDT1u');
Z('HS2_1s') -> Y('HDT1u');
Y('HDT1u') -> Z('H2S1s') or Z('LPG1s') or Z('LSRN2s') or Z('HSRN3s') or
Z('NAP4s') or Z('FG1s');
Z('H2S1s') -> Y('HDT1u');
Z('LPG1s') -> Y('HDT1u');
Z('LSRN2s') -> Y('HDT1u');
Z('HSRN3s') -> Y('HDT1u');
Z('NAP4s') -> Y('HDT1u');
Z('FG1s') -> Y('HDT1u');

*HDT 2
Y('HDT2u') -> Z('NAP2s') or Z('H2_2s');
Z('NAP2s') -> Y('HDT2u');
Z('H2_2s') -> Y('HDT2u');
Y('HDT2u') -> Z('H2S2s') or Z('LPG3s') or Z('LSRN3s') or Z('HSRN4s') or
Z('NAP3s') or Z('FG2s');
Z('H2S2s') -> Y('HDT2u');
Z('LPG3s') -> Y('HDT2u');
Z('LSRN3s') -> Y('HDT2u');
Z('HSRN4s') -> Y('HDT2u');
Z('NAP3s') -> Y('HDT2u');
Z('FG2s') -> Y('HDT2u');

*ISO
Z('LSRN5s') -> Y('ISOu');
Y('ISOu') -> Z('LSRN5s');
Y('ISOu') -> Z('ISOs') or Z('FG4s');
Z('ISOs') -> Y('ISOu');
Z('FG4s') -> Y('ISOu');

*SRU
Y('SRUu') -> Z('H2S1s') or Z('H2S2s');
Z('H2S1s') -> Y('SRUu');
Z('H2S2s') -> Y('SRUu');
Y('SRUu') -> Z('H2S1s');
Y('SRUu') -> Z('H2S2s');
Y('SRUu') -> Z('Ss') or Z('TGs');
Z('Ss') -> Y('SRUu');
Z('TGs') -> Y('SRUu');

*REF
Y('REFu') -> Z('HSRN5s') or Z('NAP5s');
Z('HSRN5s') -> Y('REFu');
Z('NAP5s') -> Y('REFu');
Y('REFu') -> Z('H2s') or Z('FG3s') or Z('LPG2s') or Z('REFs');
Z('H2s') -> Y('REFu');
Z('FG3s') -> Y('REFu');
Z('LPG2s') -> Y('REFu');
Z('REFs') -> Y('REFu');

*SOLD
Y('SOLDu') -> Z('LSRN6s') or Z('GSLNs') or Z('Ss') or Z('LPG5s');
Z('LSRN6s') -> Y('SOLDu');
Z('GSLNs') -> Y('SOLDu');
Z('Ss') -> Y('SOLDu');
Z('LPG5s') -> Y('SOLDu');
Z('SOLDs') -> Y('SOLDu');
Z('SOLDs') -> Y('SOLDu');

*BLND
Y('BLNDu') -> Z('ISOs') or Z('REFs');
Z('ISOs') -> Y('BLNDu');
Z('REFs') -> Y('BLNDu');
Y('BLNDu') -> Z('GSLNs');
Z('GSLNs') -> Y('BLNDu');

*LPG
Z('LPG4s') -> Y('LPGu');
Y('LPGu') -> Z('LPG4s');
Y('LPGu') -> Z('LPG5s');
Z('LPG5s') -> Y('LPGu');

*FGH
Y('FGHu') -> Z('FG1s') or Z('FG2s') or Z('FG3s') or Z('FG4s');
Z('FG1s') -> Y('FGHu');
Z('FG2s') -> Y('FGHu');
Z('FG3s') -> Y('FGHu');
Z('FG4s') -> Y('FGHu');
Y('FGHu') -> Z('FG5s');
Z('FG5s') -> Y('FGHu');

*SPLT 1
Z('LSRN4s') -> Y('SPLT1u');
Y('SPLT1u') -> Z('LSRN4s');
Y('SPLT1u') -> Z('LSRN5s') or Z('LSRN6s');
Z('LSRN5s') -> Y('SPLT1u');
Z('LSRN6s') -> Y('SPLT1u');

*SPLT 2
Z('H2s') -> Y('SPLT2u');
```

```
Y('SPLT2u') -> Z('H2s');
Y('SPLT2u') -> Z('H2_1s') or Z('H2_2s');
Z('H2_1s') -> Y('SPLT2u');
Z('H2_2s') -> Y('SPLT2u');

*MIX 1
Z('HSRN1s') or Z('VIS_1s') or Z('COK_1s') or Z('FCC_1s') or Z('HCR_1s') or
Z('PCHN1_1s') -> Y('MIX1u');
Z('HSRN1s') -> Y('MIX1u');
Z('VIS_1s') -> Y('MIX1u');
Z('COK_1s') -> Y('MIX1u');
Z('FCC_1s') -> Y('MIX1u');
Z('HCR_1s') -> Y('MIX1u');
Z('PCHN1_1s') -> Y('MIX1u');
Y('MIX1u') -> Z('HSRN2s');
Z('HSRN2s') -> Y('MIX1u');

*MIX 2
Z('NAP1s') or Z('VIS_2s') or Z('COK_2s') or Z('FCC_2s') or Z('HCR_2s') or
Z('PCHN1_2s') -> Y('MIX2u');
Z('NAP1s') -> Y('MIX2u');
Z('VIS_2s') -> Y('MIX2u');
Z('COK_2s') -> Y('MIX2u');
Z('FCC_2s') -> Y('MIX2u');
Z('HCR_2s') -> Y('MIX2u');
Z('PCHN1_2s') -> Y('MIX2u');
Y('MIX2u') -> Z('NAP2s');
Z('NAP2s') -> Y('MIX2u');

*MIX 3
Z('LSRN1s') or Z('PCHN2s') or Z('LSRN2s') or Z('LSRN3s') -> Y('MIX3u');
Z('LSRN1s') -> Y('MIX3u');
Z('PCHN2s') -> Y('MIX3u');
Z('LSRN2s') -> Y('MIX3u');
Z('LSRN3s') -> Y('MIX3u');
Y('MIX3u') -> Z('LSRN4s');
Z('LSRN4s') -> Y('MIX3u');

*MIX 4
Z('HSRN3s') or Z('HCR_3s') or Z('PCHN3_1s') or Z('HSRN4s') -> Y('MIX4u');
Z('HSRN3s') -> Y('MIX4u');
Z('HCR_3s') -> Y('MIX4u');
Z('PCHN3_1s') -> Y('MIX4u');
Z('HSRN4s') -> Y('MIX4u');
Y('MIX4u') -> Z('HSRN5s');
Z('HSRN5s') -> Y('MIX4u');

*MIX 5
Z('NAP3s') or Z('NAP4s') or Z('HCR_4s') or Z('PCHN3_2s') -> Y('MIX5u');
Z('NAP3s') -> Y('MIX5u');
Z('NAP4s') -> Y('MIX5u');
Z('HCR_4s') -> Y('MIX5u');
Z('PCHN3_2s') -> Y('MIX5u');
Y('MIX5u') -> Z('NAP5s');
Z('NAP5s') -> Y('MIX5u');

*MIX 6
Z('LPG1s') or Z('LPG2s') or Z('LPG3s') -> Y('MIX6u');
Z('LPG1s') -> Y('MIX6u');
Z('LPG2s') -> Y('MIX6u');
Z('LPG3s') -> Y('MIX6u');
Y('MIX6u') -> Z('LPG4s');
Z('LPG4s') -> Y('MIX6u');

*VIS
Y('VISu') -> Z('VIS_1s') or Z('VIS_2s');
Z('VIS_1s') -> Y('VISu');
Z('VIS_2s') -> Y('VISu');

*COK
Y('COKu') -> Z('COK_1s') or Z('COK_2s');
Z('COK_1s') -> Y('COKu');
Z('COK_2s') -> Y('COKu');

*FCC
Y('FCCu') -> Z('FCC_1s') or Z('FCC_2s');
Z('FCC_1s') -> Y('FCCu');
Z('FCC_2s') -> Y('FCCu');

*HCR
Y('HCRu') -> Z('HCR_1s') or Z('HCR_2s') or Z('HCR_3s') or Z('HCR_4s');
Z('HCR_1s') -> Y('HCRu');
Z('HCR_2s') -> Y('HCRu');
Z('HCR_3s') -> Y('HCRu');
Z('HCR_4s') -> Y('HCRu');

$OFFECHO
*END LOGMIP SECTION

OPTION
MIP = LMCHULL
LIMROW = 10000
LIMCOL = 10000

;

MODEL
naphtha_opt_heavy
/ALL/;

SOLVE
naphtha_opt_heavy using MIP minimizing COST;

DISPLAY
COST.l,f.l,y.l,z.l;
```

## APPENDIX C

## MILP GAMS RESULT (HEAVY CRUDE CHARGE)

---- 1591 VARIABLE c.L = 2743.055 total cost of refinery

---- 1591 VARIABLE F.L stream flowrates

| Stream | Flowrate (kg/d) | Stream | Flow rate (kg/d) | Stream | Flow rate (kg/d) |
|---|---|---|---|---|---|
| COK_1 | 2000000 | HCR_4 | 0 | NAP1 | 0 |
| COK_2 | 0 | HSRN1 | 5029419 | NAP2 | 0 |
| FCC_1 | 2000000 | HSRN2 | 1.1E+7 | NAP3 | 0 |
| FCC_2 | 0 | HSRN3 | 0 | NAP4 | 2.3E+7 |
| FG1 | 128283 | HSRN4 | 0 | NAP5 | 2.3E+7 |
| FG2 | 0 | HSRN5 | 0 | PCHN1_1 | 0 |
| FG3 | 855328 | ISO | 1622356 | PCHN1_2 | 0 |
| FG4 | 16387 | LPG1 | 68261 | PCHN2 | 0 |
| FG5 | 1000000 | LPG2 | 1803125 | PCHN3_1 | 0 |
| GSLN | 2.1E+7 | LPG3 | | PCHN3_2 | 0 |
| H2 | 739743 | LPG4 | 1871386 | REF | 1.9E+7 |
| H2_1 | 739743 | LPG5 | 1871386 | S | 11973 |
| H2_2 | 0 | LSRN1 | 1820827 | SOLD | 2.3E+7 |
| H2S1 | 14122 | LSRN2 | 0 | TG | 2149 |
| H2S2 | 0 | LSRN3 | 0 | VIS_1 | 0 |
| HCR_1 | 2000000 | LSRN4 | 1820827 | VIS_2 | 0 |
| HCR_2 | 0 | LSRN5 | 1638744 | CR | 3.2E+7 |
| HCR_3 | 0 | LSRN6 | 182082 | | 0 |

---- 1591 VARIABLE Y.L

ADUu 1.000, BLNDu 1.000, COKu 1.000, FCCu 1.000, FGHu 1.000
HCRu 1.000, HDT1u 1.000, ISOu 1.000, LPGu 1.000, MIX1u 1.000
MIX3u 1.000, MIX5u 1.000, MIX6u 1.000, REFu 1.000, SPLT1u 1.000
SPLT2u 1.000, SRUu 1.000, SOLDu 1.000

---- 1591 VARIABLE Z.L

COK_1s 1.000, CRs 1.000, FCC_1s 1.000, FG1s 1.000, FG3s 1.000
FG4s 1.000, FG5s 1.000, GSLNs 1.000, H2s 1.000, H2_1s 1.000
H2S1s 1.000, HCR_1s 1.000, HSRN1s 1.000, HSRN2s 1.000, ISOs 1.000
LPG1s 1.000, LPG2s 1.000, LPG4s 1.000, LPG5s 1.000, LSRN1s 1.000
LSRN4s 1.000, LSRN5s 1.000, LSRN6s 1.000, NAP4s 1.000, NAP5s 1.000
REFs 1.000, Ss 1.000, SOLDs 1.000, TGs 1.000

EXECUTION TIME = 0.053 SECONDS 3 Mb WIN228-228 Jul 26, 2008

# APPENDIX D

## GDP GAMS RESULT (HEAVY CRUDE CHARGE)

---- 1591 VARIABLE c.L = 2743.055 total cost of refinery

---- 1591 VARIABLE F.L stream flowrates

| Stream | Flowrate (kg/d) | Stream | Flow rate (kg/d) | Stream | Flow rate (kg/d) |
|---|---|---|---|---|---|
| COK_1 | 2000000 | HCR_4 | 0 | NAP1 | 0 |
| COK_2 | 0 | HSRN1 | 5029521 | NAP2 | 0 |
| FCC_1 | 2000000 | HSRN2 | 1.1E+7 | NAP3 | 0 |
| FCC_2 | 0 | HSRN3 | 0 | NAP4 | 2.4E+7 |
| FG1 | 128194 | HSRN4 | 0 | NAP5 | 2.4E+7 |
| FG2 | 0 | HSRN5 | 0 | PCHN1_1 | 0 |
| FG3 | 855419 | ISO | 1622447 | PCHN1_2 | 0 |
| FG4 | 16478 | LPG1 | 68352 | PCHN2 | 0 |
| FG5 | 1000000 | LPG2 | 1803216 | PCHN3_1 | 0 |
| GSLN | 2.2E+7 | LPG3 | | PCHN3_2 | 0 |
| H2 | 739834 | LPG4 | 1871477 | REF | 1.9E+7 |
| H2_1 | 739834 | LPG5 | 1871477 | S | 11164 |
| H2_2 | 0 | LSRN1 | 1820918 | SOLD | 2.4E+7 |
| H2S1 | 14223 | LSRN2 | 0 | TG | 2231 |
| H2S2 | 0 | LSRN3 | 0 | VIS_1 | 0 |
| HCR_1 | 2000000 | LSRN4 | 1820918 | VIS_2 | 0 |
| HCR_2 | 0 | LSRN5 | 1638835 | CR | 3.3E+7 |
| HCR_3 | 0 | LSRN6 | 182173 | | 0 |

---- 1591 VARIABLE Y.L

ADUu 1.000, BLNDu 1.000, COKu 1.000, FCCu 1.000, FGHu 1.000
HCRu 1.000, HDT1u 1.000, ISOu 1.000, LPGu 1.000, MIX1u 1.000
MIX3u 1.000, MIX5u 1.000, MIX6u 1.000, REFu 1.000, SPLT1u 1.000
SPLT2u 1.000, SRUu 1.000, SOLDu 1.000

---- 1591 VARIABLE Z.L

COK_1s 1.000, CRs 1.000, FCC_1s 1.000, FG1s 1.000, FG3s 1.000
FG4s 1.000, FG5s 1.000, GSLNs 1.000, H2s 1.000, H2_1s 1.000
H2S1s 1.000, HCR_1s 1.000, HSRN1s 1.000, HSRN2s 1.000, ISOs 1.000
LPG1s 1.000, LPG2s 1.000, LPG4s 1.000, LPG5s 1.000, LSRN1s 1.000
LSRN4s 1.000, LSRN5s 1.000, LSRN6s 1.000, NAP4s 1.000, NAP5s 1.000
REFs 1.000, Ss 1.000, SOLDs 1.000, TGs 1.000

EXECUTION TIME = 0.017 SECONDS 3 Mb WIN228-228 Jul 26, 2008